In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

MANET Global Connectivity and Mobility Management Using HMIPv6 and OLSR

By

Maoning Wang

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science

School of Computer Science
Carleton University
Ottawa, Ontario

August 22, 2003

© Copyright
2003, Maoning Wang

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88974-2

Our file Notre référence

ISBN: 0-612-88974-2

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

Abstract

To achieve the seamless mobility for Internet mobile users who roam between Mobile Ad-hoc Networks (MANETs) and other parts of the Internet, one must consider a mechanism for efficient mobility management and handoff support in such an environment. This thesis presents a novel scheme to integrate MANETs with the Internet and support mobility across Wireless Local Area Networks (WLANs) and MANETs. The mobiles, connected as a MANET, employ the Optimized Link State Routing (OLSR) protocol for routing within the MANET. Mobility management across WLANs and MANETs is achieved through the Hierarchical Mobile IPv6 (HMIPv6) protocol. To the best of our knowledge, this is the first proposal to exploit the salient features of HMIPv6 in such a hybrid network architecture integrating MANETs with the Internet.

An implementation of the OLSR with HMIPv6 scheme is also presented in this thesis, which extends the OLSR protocol and ICMP6 protocol to support the cooperation of OLSR and HMIPv6.

Another contribution of this thesis is a performance evaluation on a HMIPv6 based test-bed composed of WLANs and MANETs. A set of performance benchmarks and metrics is exploited to quantify the impact of various factors of OLSR and HMIPv6 on the handoff latency and packet loss. Furthermore, the efficiency gain obtained from using HMIPv6 in such a hybrid network is investigated. The investigation result shows that the use of HMIPv6 can achieve up to 27% gain on reducing the handoff latency when a mobile roams within a domain. Concerning the reduction of the signaling load on the Internet, the use of HMIPv6 can achieve at least a 54% gain and converges to 69%.

Acknowledgement

First and foremost, my grateful thanks go to Dr. Michel Barbeau, the supervisor of my Master thesis. His invaluable and patient guidance, enthusiastic encourage and support accompany me in every stage of my thesis research. What I learn from him will provide me with lifetime benefits. I consider myself luck to access his supervision.

I would like to thank Ms. Louise Lamont, the project leader in VPNT group in Communication Research Center of Canada (CRC), for her comprehensive supervision and great support on the work of this thesis. The financial support of this project from CRC is gratefully acknowledged. I am grateful to Dr. Thomas Kunz for his valuable suggestions on my thesis work.

Finally, I want to express my deepest feeling to my family. My special thanks go to my parents, Mr. Tianwen Wang and Mrs. DaFeng Ye. Without their love and encouragement, I would not pursue new goals again and again. I especially thank my sister, Ms. Maoyu Wang, and my brother-in-law, Mr. Tao Chen. They provided me great support and help during my study. I also thank my nephew, lovely little Yuang, whose brightness and happiness always fresh my mind. Especially, I give my thanks to Mr. Ning Yang for his deep love and encouragement.

Table of Contents

Abstract	III
Acknowledgement.....	IV
Table of Contents	V
List of Figures	VIII
List of Tables.....	X
List of Acronyms.....	XI
 Chapter 1 Introduction.....	 1
1.1 Context of the work.....	1
1.2 Motivation	3
1.3 Contribution	5
1.4 Road map.....	6
 Chapter 2 Integration of MANETs and the Internet	 8
2.1 Mobility management in the Internet	8
2.1.1 Mobile IP.....	9
2.1.2 Mobility support in IPv6	12
2.2 MANET and its routing protocols.....	13
2.2.1 Table-driven routing protocols.....	14
2.2.2 On-demand routing protocols.....	19
2.3 Network architecture for integrating a MANET with the Internet.....	23
2.4 Existing proposals for integrating MANET routing with Mobile IP	26
2.4.1 DSR with Mobile IP	26
2.4.2 MIPMANET.....	28
2.4.3 Global connectivity for IPv6 MANETs	30
2.4.4 RIP with Mobile IP	33
2.4.5 MEWLANA-TD	36
2.4.6 CRC OLSR with Mobile IPv6	38
2.5 Performance analysis of existing integration methods.....	40

Chapter 3	OLSR and HMIPv6 Review.....	42
3.1	HMIPv6.....	42
3.1.1	HMIPv6 overview	43
3.1.2	HMIPv6 operation.....	45
3.1.3	Performance comparison of basic Mobility support in IPv6 and HMIPv6 ...	50
3.2	OLSR protocol	54
3.2.1	OLSR overview.....	54
3.2.2	OLSR operation.....	55
3.2.3	OLSR networking support	57
Chapter 4	Design of an OLSR with HMIPv6 Network	60
4.1	Network architecture	61
4.2	CoA autoconfiguration technique	61
4.2.1	RA approach.....	63
4.2.2	ACM approach and design decision.....	64
4.3	Join MANET mechanism.....	65
4.3.1	Using home address	66
4.3.2	Using RCoA	67
4.3.3	Using LCoA	68
4.3.4	Design decision of joining MANET mechanism	68
4.4	New message structure.....	69
4.4.1	ACM message	70
4.4.2	Local RA message.....	70
4.5	Regional mobility	71
4.6	Local mobility	73
4.7	Data transfer	74
4.8	Message sequence diagram	75
4.8.1	Message sequence diagram in handoff process.....	75
4.8.2	Message sequence diagram in data transfer process	76
Chapter 5	OLSR with HMIPv6 Implementation	79

5.1	Implementation environment	80
5.2	Architecture of OLSR with HMIPv6 implementation	81
5.3	Communication between user space and kernel space.....	83
5.3.1	Communication from user space to kernel space.....	84
5.3.2	Communication from kernel space to user space.....	85
5.4	IPC in user space	87
5.5	MN implementation	88
5.5.1	Architecture.....	88
5.5.2	Local RA receiver	90
5.6	GW implementation	92
Chapter 6	Experiments and Evaluation	94
6.1	Testbed	94
6.2	Functionality test.....	96
6.2.1	Test cases.....	97
6.2.2	Test results.....	97
6.3	Performance test.....	99
6.3.1	Test metrics	99
6.3.2	Test scenarios	105
6.3.3	Handoff time and packet loss.....	106
6.3.4	Signaling load.....	114
6.4	Evaluation of gain	114
6.4.1	Gain on handoff latency	115
6.4.2	Gain on signaling load.....	116
Chapter 7	Conclusion and Future Work	118
7.1	Conclusion.....	118
7.2	Future work	122
Appendix	Control Flows of OLSR with HMIPv6 Implementation.....	128

List of Figures

Figure 2.1: Internet-MANET hybrid network.....	24
Figure 2.2: Route discovery for node D on the Internet.....	28
Figure 2.3: Software architecture of RIP with Mobile IP	35
Figure 2.4: Network architecture of MEWLANA-TD.....	36
Figure 2.5: WLANs & MANETs based Mobile network [15]	38
Figure 3.1: Hierarchical tree structure in HMIPv6	44
Figure 3.2: Regional mobility in HMIPv6	47
Figure 3.3: Local mobility in HMIPv6.....	48
Figure 3.4: Data transfer in HMIPv6	49
Figure 3.5: Studied scenarios	51
Figure 4.1: Architecture of an OLSR with HMIPv6 network.....	62
Figure 4.2: ACM message data structure	70
Figure 4.3: Local RA message data structure	71
Figure 4.4: Regional mobility in OLSR with HMIPv6.....	72
Figure 4.5: Local mobility in OLSR with HMIPv6	74
Figure 4.6: Data transfer in OLSR with HMIPv6	75
Figure 4.7: Message diagram in handoff process.....	77
Figure 4.8: Message diagram in data transfer process	78
Figure 5.1: Architecture of GW implementation	82
Figure 5.2: Architecture of MN implementation	83
Figure 5.3: Communication from user space to kernel space on MN.....	85
Figure 5.4: Communication from kernel space to user space on an MN	86
Figure 5.5: MN software architecture	89
Figure 5.6: Software structure on GW	92
Figure 6.1: OLSR with HMIPv6 testbed.....	96
Figure 6.2: WLAN→MANET local handoff latency decomposition.....	103
Figure 6.3: WLAN→MANET regional handoff latency decomposition	104
Figure 6.4: Decomposition of handoff latency (Hop cnt. = 1).....	109

Figure 6.5: Decomposition of handoff latency (Hop cnt. = 2).....	109
Figure 6.6: Handoff latency vs ACM interval.....	111
Figure 6.7: Handoff latency vs. WAN channel delay (Hop cnt. = 1).....	112
Figure 6.8: Handoff latency vs. WAN channel delay (Hop cnt. = 2).....	113
Figure 6.9: $G_{SIG,AV}$ for different values of N_{CN}	117
Figure A.1: Control Flow in User Space on MN	129
Figure A.2: Control Flow in CRC OLSR6 on MN	130
Figure A.3: Control Flow in RADVD-CRC-mn on MN	131
Figure A.4: Control Flow in Kernel Space on MN	132
Figure A.5: Control Flow of Local Handoff on MN.....	133
Figure A.6: Control Flow of Regional Handoff on MN	134
Figure A.7: Control Flow in RADVD-CRC-gw on GW	135
Figure A.8: Control Flow in CRC OLSR6 on GW	136

List of Tables

Table 2.1: DSDV operations	17
Table 2.2: WRP operations	18
Table 2.3: DSR route discovery operations	21
Table 2.4: AODV route discovery operations.....	22
Table 3.1: Control messages sent and received by an MN in a handoff	53
Table 6.1: Local handoff latency and packet loss	107
Table 6.2: Regional handoff latency and packet loss.....	108
Table 6.3: Control messages sent and received by an MN per handoff.....	114
Table 6.4: G_{LO} in local handoff.....	115
Table 6.5: G_{REG} in regional handoff.....	115

List of Acronyms

ACM	Auto Configuration Message
CoA	Care-of Address
CN	Correspond Node
CRC	Communication Research Center of Canada
DSR	Dynamic Source Routing
HA	Home Agent
HMIPv6	Hierarchical Mobile IPv6
IPC	Inter-process Communication
LCoA	Local Care-of Address
MN	Mobile Node
MANET	Mobile Ad Hoc Network
MAP	Mobile Anchor Point
MIPMANET	Mobile IP for Mobile Ad Hoc Network
MEWLANA-TD	Mobile Enriched Wireless Local Area Network Architecture
OLSR	Optimized Link State Routing
RCoA	Regional Care-of Address
RA	Route Advertisement
RT	Route Table
RIP	Routing Information Protocol
WLAN	Wireless Local Area Network
WAN	Wide Area Network

Chapter 1 Introduction

1.1 Context of the work

During the last several years a trend towards using IP as the dominant network protocol is becoming stronger. All-IP based networks are being designed and will soon be widely available. With the popular use of mobile devices, the desire for mobile access to the Internet is increasing. An important area of IP research is mobile wireless networking.

There are currently two variations of mobile wireless networks. The first type of mobile wireless network is known as an infrastructure network, i.e., a network with fixed and wired gateways. A mobile unit within these networks connects to the nearest base station that is within its communication radius. As the mobile device travels out of range of one base station and into the range of another, a “handoff” occurs from the previous base station to the new one, and the mobile device can continue communication seamlessly throughout the network. Typical applications of this type of network include wireless local area networks (WLANs).

The second type of wireless network, which lacks an underlying infrastructure mobile network, is commonly known as a mobile ad-hoc network (MANET). MANETs have no fixed infrastructure, and all nodes are capable of movement and can be connected dynamically in an arbitrary manner. Nodes of these networks can serve as routers that discover and maintain routes to other nodes in the network. Some examples of the possible uses of MANET include participants sharing information in a meeting, soldiers relaying information for situation awareness on the battlefield, and emergency disaster relief personnel coordinating efforts after a hurricane or earthquake [12].

There have been several proposals for IP-based mobility management protocols. The Mobile IPv4 and Mobility support in IPv6 proposed by the IETF working group on Mobile IP are two main protocols for supporting IP mobility. However, in some environments where mobile hosts change their point of attachment to the network very frequently, the basic Mobile IP protocols could introduce significant network overhead in terms of increased delay, packet loss and signaling [1]. The overhead may degrade the service of many real-time wireless applications, such as voice-over-IP. As a result, a number of IP micro-mobility protocols have been proposed to handle local movement (e.g. within a domain) of mobile hosts. The main IP micro-mobility protocols include Cellular IP [30], Hawaii [28] and Hierarchical Mobile IP [7]. The Cellular IP protocol supports paging and a number of handoff techniques. In Cellular IP, paging caches coarsely maintain the position of idle mobile hosts in a service area. Using the paging caches, Cellular IP can efficiently point out idle mobile hosts that wish to engage in active communications. The mobile hosts do not need to update their location after each local handoff. Those mobile hosts currently receiving and expecting to receive data also maintain another set of mappings called a routing cache. The distributed routing cache maintains the positions of active mobile hosts in the service area and dynamically refreshes the routing state in response to the handoff of active mobile hosts.

Hawaii uses a separate routing protocol to handle intra-domain mobility and relies on Mobile IP to handle inter-domain mobility. In a foreign domain, a mobile node retains a co-located care-of address unchanged so that its movements within a domain are transparent to its home agent. To handle the movement within a domain, Hawaii nodes not only act as a classical IP router but also maintain routing caches to manage the

mobility. The mobility-specific routing information in a routing cache is created, updated and modified by explicit signaling messages sent by mobile nodes.

The Hierarchical Mobile IP protocol employs a hierarchy of foreign agents to handle local mobility. Home agents outside the domain only know inter-domain movements of mobile hosts. When a mobile host does the handoff between different access networks in the same domain, it only registers its new location with a gateway foreign agent in the domain. Tunnels connecting the gateway foreign agent to foreign agents located at access points are exploited to transfer packets from external hosts to the mobile host. Local mobility is managed locally and transparently to the domain's external hosts. Compared to basic Mobile IP, Hierarchical Mobile IP can reduce both the delay during local handoff and the signaling load experienced by the network by eliminating registration between mobile hosts and possibly remote home agents when mobile hosts move within the same domain. In this thesis, Hierarchical Mobile IP v6 protocol is used to manage the micro-mobility.

1.2 Motivation

MANETs are often put into the Internet to fill the coverage gap where WLANs are not available. Since nodes in MANETs are inherently mobile, some of these mobile nodes roam between MANETs and other parts of the Internet. To achieve seamless mobility for Internet mobile users who travel across multiple WLANs and MANETs, one must consider a mechanism for efficient mobility management and handoff support in such an environment. Recent research has focused on achieving this purpose. In the network architecture presented by the existing integration methods, a foreign agent or a gateway

serves as a bridge between a MANET and the Internet to provide Internet access and Mobile IP service for mobile hosts in the MANET. The route advertisement or agent advertisement mechanism in Mobile IP is limited to nodes within the single communication hop of an FA or gateway. The existing integration methods have to extend the ICMP advertisement or introduce a new mechanism to broadcast the existence of a foreign agent or gateway to MANET nodes, which may be multiple hops away from the foreign agent or gateway.

Most of the research on this topic concentrates on on-demand MANET routing protocols, including the integration of AODV (Ad hoc On-Demand Distance Vector) routing protocol with Mobile IP presented in [31], MIPMANET [11] as well as the integration of the DSR (Dynamic Source Routing) protocol with Mobile IP [12], etc. On the other hand, some researchers explore the use of table-driven MANET routing protocols as a way of integrating MANETs with the Internet. MEWLANA-TD proposed in [6] employs a table-driven MANET routing protocol, the DSDV (Destination Sequenced Distance Vector) protocol. An integration approach of combining Mobile IP with the Optimized Link State Routing (OLSR) protocol is presented in [15]. The two types of integration methods have different performance features. The methods based on table-driven routing protocols could generate high ad hoc routing overhead because of routing information exchanged periodically among nodes in a MANET. However, the on-demand feature of those proposals presented in [31] [11] [12] could result in longer handoff latency than the table-driven integration methods.

Given the existing ways of integrating MANETs with the Internet, one of the objectives of this thesis is to explore the following problems: Compared with most existing

integration approaches based on on-demand routing protocol, is it worthwhile to exploit a table-driven MANET routing protocol to facilitate the integration of MANETs with the Internet? How can handoff latency be reduced without triggering high ad hoc routing overhead in the hybrid network environment?

Another objective of our work is to reconsider the use of Mobile IP or mobility management in IPv6 in the existing integration proposals. In such an environment where MNs roam frequently across MANETs and WLANs within the same domain, the use of Mobile IP or mobility management in IPv6 could result in long handoff latency and high signaling load over the Internet. Hierarchical Mobile IPv6 (HMIPv6), an extension of the basic Mobility support in IPv6, is presented in [3] to handle micro-mobility. Can we employ HMIPv6 to manage the mobility between WLANs and MANETs? To integrate HMIPv6 protocol with a MANET routing protocol, what extensions and modifications need to be done in HMIPv6 and the routing protocol? How much efficiency gain is obtained in the use of HMIPv6? The thesis focuses on these problems.

1.3 Contribution

In this thesis, we present a novel scheme to integrate MANETs with the Internet and support mobility across WLANs and MANETs. To the best of our knowledge, this is the first proposal to exploit the salient features of HMIPv6 in a hybrid network architecture of integrating MANETs with the Internet. The main contributions of this thesis are the following:

- Design of a novel scheme to integrate MANETs with the HMIPv6 based Internet.

- Extension of the OLSR protocol and ICMPv6 protocol to support the novel integration scheme.
- Implementation of the integration scheme based on the CRC OLSR6 implementation [15] and Monash's HMIPv6 implementation [4].
- Design and construction of a test-bed to verify the functionality of the scheme.
- Performance evaluation of the scheme. A set of performance benchmarks are designed and exploited to assess the effect of various factors on the integration scheme.
- Investigation of efficiency gains obtained in the use of HMIPv6 over basic MIPv6 in the integration scheme.

1.4 Road map

This thesis is organized into seven parts. In Chapter 2, we first review basic mobility management proposals employed in the Internet and MANETs, and then investigate existing approaches to integrate MANETs with the Internet. The generic network architecture for the MANET-Internet hybrid network is also given in this chapter. Chapter 3 discusses the highlights of HMIPv6 and OLSR protocols. In Chapter 4, we present our scheme of supporting the integration of MANETs and the Internet, emphasizing the problems, challenges and our solution. In Chapter 5, an implementation of our approach based on Linux is given. The features and experience of our implementation is discussed. In Chapter 6, we describe the test-bed, constructed to evaluate the performance of the implementation of our approach, and analyze the test

results in various test cases. The effects of different factors on the integration scheme are assessed according to a set of test metrics. Finally, we present our conclusions and discuss future work in Chapter 7.

Chapter 2 Integration of MANETs and the Internet

Much research has been done in order to support the integration of MANETs and the Internet. In this chapter, we first review basic mobility management proposals employed on the Internet. Mobile IP and Mobility support in IPv6 are introduced in Section 2.1. In Section 2.2, we discuss some representative MANET routing protocols, which are employed in the existing proposals of integrating MANETs with the Internet. Then, we give the generic network architecture for the MANET-Internet hybrid network in Section 2.3. In Section 2.4, most of existing proposals on integrating MANET routing protocols with Mobile IP are discussed.

2.1 Mobility management in the Internet

The increasing variety of wireless devices, such as PDAs, handhelds and digital cellular phones, is generating many requests for mobility management in the Internet. A mobile user prefers that communication with other Internet users not be disrupted when the user changes the mobile devices' point of attachment to the Internet. The fundamental routing technique on the Internet, provided by the Internet Protocol, routes packets to their destinations according to IP addresses. These addresses are associated with a fixed network location and contain a network number. Hence, when a mobile node (MN) moves to a new network, its previous IP address does not indicate its new location, making transparent mobility impossible.

Mobile IP (RFC 2002) [21] is a standard proposed by IETF, which is designed to solve this problem by using two IP addresses for an MN: a fixed home address to identify the

MN, and a changeable care-of-address (CoA) to associate with the MN's current location. In this section, we review the mobility management technique in Mobile IP. Additionally, we describe how Mobile IP changes with IPv6 [10].

2.1.1 Mobile IP

In Mobile IP, the home address is static and serves as the identifier of an MN. The CoA changes at each new point of attachment and can be thought of as the MN's topologically significant address. To exploit the two addresses to manage an MN's movement, Mobile IP introduces two agents: Home Agent (HA) and Foreign Agent (FA). The HA can be viewed as a proxy of an MN in the home network. When the MN moves out of the home network, the HA can receive all packets destined for the MN's home address and arranges to deliver them to the MN's current point of attachment. The FA is a host in the foreign network which the MN is visiting, and can provide an access point for the MN and deliver information between an MN and the HA.

Mobile IP can be understood as the cooperation of three separable mechanisms: Discovering the CoA; Registering the CoA; Tunneling to the CoA [22]

- Discovering the CoA

In Mobile IP, an existing standard Router Advertisement (RA) is extended to support the discovery process of CoAs. The ICMP RA is modified to carry the necessary CoAs except for the default router information, which is known as an Agent Advertisement. The HA and FAs broadcast Agent Advertisements periodically. An HA uses the Agent Advertisement to make itself known while it does not offer any CoA. In an Agent Advertisement from a FA, there are one or more available CoAs listed. If an MN needs to get a CoA and would not like to wait for the periodic advertisement, the MN can

broadcast an Agent Solicitation. Any FA or HA that receives the Agent Solicitation can reply to the MN by unicasting an Agent Advertisement to it.

By receiving Agent Advertisement messages, an MN can do three things. First, the MN can perform movement detection by comparing the network prefix contained in an Agent Advertisement with its previous network prefix. If the network prefix is its home network's prefix, the MN learns that it is in the home network. If the network prefix is different than the previous one, the MN detects that it has arrived in a new network. Secondly, the MN can discern special features provided by FAs from Agent Advertisements. Finally, the MN can select a CoA from the CoAs list in the Agent Advertisement to identify its current attachment point.

- Registering the CoA

After discovering a new CoA, an MN needs a registration to its HA to apply the HA's service. The MN registers its new CoA with its HA by sending out Registration Requests. Registration Requests are sent first to an FA in the foreign network where the MN is visiting. The FA records the MN's information, including its home address and MAC address, in a visitor list. Then, the FA relays the Registration Requests to the MN's HA. Upon reception of a Registration Request, the HA adds the necessary information to its route table and builds a binding containing the MN's home address, CoA and the registration lifetime for the MN. Then, the FA sends a Registration Reply back to the MN, which includes the registration lifetime. Before the lifetime is expired, the MN must resend a Registration Request to the HA.

A Registration Request can be considered a binding update sent by the MN. A binding update is a remote redirection since it is sent remotely to the HA and can affect the HA's

route table. So, the HA must make sure the registration request was originated by the MN and not by a malicious node pretending to be the MN. Hence, both registration request and registration reply must be authenticated. Each MN and HA must share a security association and be able to use MD5 to create a digital signature.

- Tunneling to the CoA

After accepting a Registration Request of an MN, an HA will provide tunneling service for the data transfer from a Correspondent Node (CN) to the MN. When a packet from a CN is sent to the MN's home address, the HA intercepts the packet and encapsulates it in the MN's current CoA by IP-in-IP, GRE and minimal encapsulation. Using IP-in-IP, a tunnel IP header is added to the packet, which uses the MN's CoA as the destination address and uses the HA as the source address. When the tunneled packet is sent to an FA providing the CoA for the MN, the FA decapsulates the packet by removing the tunnel header and delivers it to the MAC of the MN.

In [21], all packets destined to an MN have to travel through the MN's HA first and then are delivered to the MN. This triangle routing is one of the main problems in IETF Mobile IP [23]. Therefore, route optimization protocol is proposed in [23]. The basic idea of the route optimization mechanism is to send binding updates directly from an MN to all of its CNs. Upon receiving a binding update message from an MN, a CN can build a binding between the MN's home address and CoA so that it sends packets directly to the MN without the service of an HA.

2.1.2 Mobility support in IPv6

IPv6 [5] includes many features missed in IPv4, which can facilitate mobility. Mobility support in IPv6 [10] follows the main design idea in Mobile IPv4, including the registration of CoA and the use of encapsulation to transfer data from an MN's home network to its current location. However, new features in IPv6 make some changes to the mobility management mechanism of Mobile IPv4. These changes include the elimination of FA, route optimization requirements for each MN, and built-in security features.

- FA not existing

The stateless address autoconfiguration and neighbor discovery features in IPv6 support movement detection and the acquiring of CoA without the need for FAs. The routers in foreign networks broadcast periodically the RA messages to declare their existence and indicate the network prefix. An MN can learn of its arrival to a new foreign network by receiving the RAs or sending the Router Solicitation (RS) messages. After receiving an RA with a new network prefix, the MN can configure its new CoA using the network prefix and its own MAC address by stateless address autoconfiguration technique. Then, the MN can register the new CoA with its HA by sending Binding Update (BU) messages. After building the binding entry for the MN, the HA can intercept data packets routed to the MN's home address and tunnel them to the MN's current CoA. As the end point of the tunneling, the MN decapsulates these packets and transfers the data to a high layer such as TCP. The whole mobility management process does not need the service of FAs.

- Route optimization

The route optimization mechanism in Mobile IPv4 allows a CN to send packets directly to an MN without the help of the HA. In the mobility support of IPv6, all nodes are required to accomplish this extra functionality. After configuring a new CoA, an MN always sends a BU option in some packets' Destination Header to all its CNs, which are recorded in a CN list. A CN can build a binding of the MN in its binding cache in response to the reception of a BU. When the CN needs to send data to the MN, it uses the Routing Header to route the packet to the MN instead of tunneling the packet.

- Security

Compared with IPv4, one of the biggest changes in IPv6 is that all nodes are expected to implement strong authentication and encryption features. So, the accomplishment of the security in Mobility support in IPv6 is simplified since all the authentication procedures have existed in IPv6.

2.2 MANET and its routing protocols

Mobile Ad Hoc network (MANET) is a type of mobile wireless networks. In contrast to an infrastructure wireless network, a MANET is an infrastructure-less network. A MANET consists of mobile nodes that connect to each other dynamically in an arbitrary fashion. In a MANET, there is no fixed router and each MN can serve as a router that discovers and maintains routes to other nodes. The MANET concept applies to situations such as emergency rescue operations and data sharing in a conference.

In contrast to wired infrastructure networks, MANETs have low bandwidth and high error rates [27]. To support the routing in the networks, many protocols have been

proposed in recent years. The MANET routing protocols can generally be categorized as on-demand routing protocols or table-driven routing protocols. In the following subsections, we review some popular MANET routing protocols and categorize them according to their characteristics.

2.2.1 Table-driven routing protocols

Table-driven routing protocols build routes in a proactive way between nodes in a MANET. Similar to the routing in infrastructure networks, route tables are created and maintained by each node to store consistent, up-to-date routing information for all other nodes. Nodes respond to the changes in the network topology by propagating update packets throughout the network. On each node, the routing information in route tables are updated and maintained according to the view of the whole network structure. The main characteristics that differentiate a table-driven routing protocol from another protocol are the number of necessary route tables and the way network topology changes are broadcast.

We introduce two well-known table-driven routing protocols as follows.

A. Destination Sequenced Distance Vector Routing

The Destination Sequenced Distance Vector Routing protocol (DSDV) [24] is a table driven algorithm based on the classical Bellman-Ford routing mechanism. The improvements made to the Bellman-Ford algorithm include the guarantee of loop-free routes and a simple route update protocol. Each routing entry in a route table is assigned a sequence number by the destination node. The sequence number enables the node to differentiate a new route from old routes so as to avoid the formation of routing loops.

When two updates have the same sequence numbers, DSDV selects the shortest path based on the hop count to the destination. On the other hand, to alleviate the traffic overhead in a MANET generated by the route update packets, DSDV uses two types of update messages. One is a *full dump*, which carries all the available routing information and is propagated infrequently during periods of occasional movement. The other is the *incremental packet* that is used to relay only the information which has changed since the last full dump. The use of small incremental packets avoids transmitting the entire route table for every change in network topology. In Table 2.1 (*refer to page 18*), we use an example to describe DSDV in operation.

DSDV requires nodes to periodically transmit route table update packets, regardless of network traffic. These update packets are broadcast throughout the network so that every node in the network knows how to reach every other node. As the number of nodes in the network grows, the size of the route tables and bandwidth required to update them also grows. This overhead is DSDV's major weakness. Furthermore, whenever the topology changes, DSDV is unstable until update packets propagate throughout the network, especially when dealing with high rates of node mobility.

B. Wireless Routing Protocol

Wireless Routing Protocol (WRP) described in [19] differs from other protocols in several ways.

Each node in the network maintains four tables: the distance table, route table, link-cost table, and message retransmission list (MRL) table. The MRL table records which updates in an update message need to be retransmitted and which neighbors should

acknowledge the retransmission [19]. In WRP, an update message is a response to link changes and is sent only between neighbor nodes. When a link between two nodes is broken, the two nodes send update messages to their neighbors. Upon reception of the update messages, these neighbor nodes update their distance tables and check for possible new paths through other nodes. The new paths are relayed back to the original nodes to update their routes. In Table 2.2 (*refer to page 19*), we use an example to describe WRP operations when a link breaks.

Except for the update messages, each node declares its existence by sending Hello messages to its neighbors during a specific period when it does not send other messages to the neighbors. If a node does not receive any message from a neighbor in a specific period, it denotes the link to the neighbor as failed.

A novel part in WRP is that it achieves freedom from routing loops. WRP avoids the “count-to-infinity” problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. This eliminates routing loop situations and provides faster route convergence when a link has failed. [29]

However, the use of four tables can lead to substantial memory requirements for nodes running WRP. Furthermore, the WRP protocol requires the transmission of hello packets whenever there is no recent transmission from a given node. The hello packets consume bandwidth and prevent a node from entering sleep mode.

<table data-bbox="233 270 428 314"><tr><th>Dest</th><th>Next hop</th><th>metric</th></tr><tr><td>A</td><td>A</td><td>1</td></tr></table>	Dest	Next hop	metric	A	A	1	Initially, the status of every link UN. Node A have a link with B.
Dest	Next hop	metric					
A	A	1
Node A moves away from B and into the vicinity of E.
A triggers an immediate incremental routing information update which was then broadcasted to D.
D determines that significant new routing information is received, so D also triggers an immediate update which carries the new routing information for node A. D would then broadcast it at every interval until the next full routing information dump.
<table data-bbox="233 1275 428 1319"><tr><th>Dest</th><th>Next hop</th><th>metric</th></tr><tr><td>A</td><td>D</td><td>3</td></tr></table>	Dest	Next hop	metric	A	D	3	Upon receiving this “incremental packet”, B updates the entry to A in its route table according to the new sequence number in the update packet. C keep the stale route to A since the update packet contains the same sequence number and metric as these in that update packet broadcasted by E.
Dest	Next hop	metric					
A	D	3					

Table 2.1: DSDV operations

<p>Initially, the status of every link. Link-costs are indicated in the figure. Node J is the destination node. The label in parentheses gives the distance and the predecessor to the destination J.</p>
<p>When link (J, K) fails, nodes J and K send update messages to their neighbors B and I.</p>
<p>Node B processes node K's update and selects link (B, J) to the destination J. Then, B relays new paths to its neighbors. When I gets node K's update, it updates its distance table entry through K and checks for the possible paths to destination J through any other neighbors. This results in the selection of the link (B, J).</p>
<p>The new path to J found by B is relayed back to the original node K. K updates its tables accordingly and reports a distance of 11 to J.</p>

Table 2.2: WRP operations

2.2.2 On-demand routing protocols

On-demand routing is the most recent entry in the class of scalable wireless routing schemes. This type of routing creates routes only when a route to the destination is needed. In a MANET running on-demand routing protocol, when an MN wants to send a packet to a destination, it initiates a route discovery process to find a route to the destination. After a route is built, it is maintained through a route maintenance process. If any link on the route is broken or the route is no longer desired, the route is deleted. Compared with table-driven routing protocols, on-demand routing protocols may have lower computation costs and lower packet overhead since they do not need to exchange routing information periodically and maintain route tables. However, when a MN using an on-demand routing protocol desires to send a packet to a destination, it has to wait until a route to the destination is discovered on-demand. This feature of on-demand routing protocols results in longer packet transfer delay than with table-driven routing protocols.

In the following, we introduce two well-known on-demand routing protocols.

A. Dynamic Source Routing

Dynamic Source Routing (DSR) is an on-demand routing protocol based on the concept of source routing, which is presented in [12]. In a source routing mechanism, each data packet carries the list of routers on the path while intermediate nodes need not keep route information.

There are two main procedures in DSR protocol: route discovery and route maintenance. Each node has a route cache storing those routes that have been found but are unexpired. When an MN needs a route to a destination, it first consults its route cache to check if

there is a route to the destination. If this route cannot be found in the route cache, the MN broadcasts a *route request* packet. Upon the reception of a *route request*, a node puts its own address in the route record of the request packet and rebroadcasts the request packet if the node does not know a route to the destination. Finally, the *route request* packet arrives at the destination or an intermediate node that knows a route to the destination. The destination or such an intermediate node will send a *route reply* packet back to the source node. To unicast the *route reply* packet, the destination node or intermediate node needs a route to the source node. There are three possible cases for attaining the reverse route. Firstly, if the destination node or intermediate node can find a route to the source node in its route cache, it can use that route. Secondly, if the route is not found in its route cache but the symmetric links are supported, the node can reverse the route in the route record of the *route request* packet. Thirdly, if the symmetric links are not supported, the node may initiate a new route discovery process to find a route to the source node. In Table 2.3, we use an example to describe the route discovery process of DSR.

To achieve route maintenance, DSR utilizes *route error* packets. When detecting a link failure, a node generates *route error* packets. When another node receives the *route error* packet, it removes the hop in error from its route cache so that all routes containing the hop are truncated at that point.

The main advantage in DSR is that it does not require the use of symmetric links between nodes. When symmetric links are not available, DSR can utilize asymmetric links.

<p>A needs a route to DST. A broadcasts a route request. The route record during route discovery is given in parentheses.</p>
<p>The intermediate nodes B and C put themselves' address to the route record and rebroadcast the request packet. We assume A and B do not know a route to DST.</p>
<p>The route request with the route record arrive DST, DST unicasts the route reply back to the source node A. In the figure, we assume the symmetric links are supported.</p>
<p>In the figure, we assume the link between DST and C is an asymmetric link. So, DST initiates a new route discovery process to find a route to the source A.</p>

Table 2.3: DSR route discovery operations

B. Ad hoc Distance Vector Routing (AODV) – The Ad hoc On Demand Distance Vector (AODV) protocol described in [25] employs a route discovery procedure similar to DSR. When a node desires to send a packet to another node in a MANET, it broadcasts a route request (RREQ) packet to its neighbors. These neighbors may return a route reply (RREP) packet back to the source node if they have a route to the destination. Otherwise, they forward the RREQ to their neighbors, and so on, until a fresh route to the destination is located.

<p>A needs a route to DST. A broadcasts a RREQ packet to its neighbors.</p>
<p>The intermediate nodes B and C record A address in their route tables, thereby build a reverse path to A. Then, B and C rebroadcast the RREQ. We assume B and C do not know a route to DST.</p>
<p>The RREQ arrives DST, DSR unicasts a RREP packet back to its neighbor C from which it first received the RREQ.</p>
<p>Upon receiving the RREP from DST, node C set up forward route entry in its route table which points to DST. Then, C routes the RREP back to the source A according to a reverse path built during the process of forwarding the RREQ.</p>

Table 2.4: AODV route discovery operations

There are a couple of important distinctions between DSR and AODV. The most notable distinction is that the AODV is a kind of hop-by-hop routing protocol in contrast to the source routing in DSR. During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby building a reverse route. If an intermediate node knows a fresh route to the destination, it unicasts a RREP to the neighbor from which it receives the RREQ. While the RREP is routed back along the reverse route, each node on the route builds a forward route entry to the destination according to the source address contained in the RREP. The different routing type makes the overhead of AODV smaller

than that of DSR since each DSR packet contains full route information, whereas in AODV packets only contain the destination address. Also, the RREP in AODV is smaller than the route reply message in DSR since the RREP only needs to carry the destination address and sequence number. In Table 2.4 (*refer to page 23*), we use an example to describe the route discovery process of AODV.

AODV is capable of both unicast and multicast routing. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees that connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. The major downside of AODV is that it requires symmetric links between nodes since the RREP is forwarded along the path established by the RREQ. Hence, AODV cannot support the use of asymmetric links.

2.3 Network architecture for integrating a MANET with the Internet

Most of the work described in Section 2.2 concentrates on stand-alone MANET networks. These protocols provide the routing support for the communication between mobile nodes within a MANET. With the growing deployment rate of wireless and mobile devices, the need for seamless roaming across WLANs and MANETs is increasing. Providing Internet connectivity for MANETs has become a hot research area. Most of the research [11], [6], [17], [31] and [26] addresses cooperation between the existing MANET routing protocols and the Mobile IP protocol. The general approach is pictured in Figure 2.1.

Figure 2.1: Internet-MANET hybrid network

To achieve the integration of MANET routing with Mobile IP, three main steps need to be accomplished: FA discovery, Registration, and Routing and addressing in a heterogeneous environment. In Mobile IP, an MN learns of the existence of an FA by receiving Agent Advertisement messages that are broadcast by the FA. In an Internet-MANET hybrid network, an MN arriving in a MANET needs to learn of the existence of the FAs to utilize their services. So, the first step in the integration is the FA discovery. When an MN learns about the presence of an FA, it selects a CoA from one of the advertised CoAs in the Agent Advertisement. Then, the MN registers its new CoA with the FA and its HA by sending a Registration Request. The Registration Request is sent to the FA using MANET routing and is then transferred to the HA through normal IP. In an infrastructure network using Mobile IP, data packets from a CN to an MN are intercepted

and tunneled to the MN's CoA. As the end point of the tunneling, an FA receives the data packets and forwards them to the MN. In the hybrid network, the FA needs to transfer the received data packets to the current location of the MN using a MANET routing.

In order to support MNs roaming in the hybrid network, two main challenges need to be faced. The first challenge is how to propagate the presence of an FA in a MANET network. Mobile IP assumes that an MN is within the transmission range of an FA. An FA broadcasts Agent Advertisement messages periodically. MNs visiting a foreign network only gain network connectivity if they receive Agent Advertisements from an FA in the network. However, in a MANET where an MN could be more than one hop away from an FA, the MN cannot receive Agent Advertisements using only a link layer broadcast scheme. A simple solution is to rebroadcast the Agent Advertisements to reach the MN outside the transmission range of the FA. However, this solution can make Agent Advertisement messages flood a MANET and generate a lot of traffic. So, some schemes are introduced to reduce this kind of Mobile IP overhead, including increasing the interval of Agent Advertisement transmission [11] and dynamic beaconing [6]. An alternative way to eliminate the overhead of Agent Advertisements is to use Agent Solicitation [17]. An FA does not advertise itself by broadcasting Agent Advertisement messages periodically, but MNs who search for an FA send Agent Solicitation messages, and FAs then unicast the Agent Advertisement messages to those MN. However, if there is a high demand for Internet connectivity by MNs in a MANET, the reduction of the overhead from Agent Advertisements may be balanced by the increase in Agent Solicitations.

The second challenge is how to utilize FAs to access the Internet for MNs in MANETs. In the hybrid network, the Internet connectivity for MANETs can be divided into two cases: outgoing packet transfer and incoming packet transfer. An outgoing packet from an MN to a node on the Internet is transferred to an FA through MANET routing and then forwarded to the destination through the infrastructure network. The various integration approaches presented in [11], [6] and [17] propose different schemes to decide whether the destination of a data packet is in the MANET or in the Internet. An incoming packet from a node on the Internet to an MN is intercepted on an FA by normal Mobile IP and transferred to the MN by MANET routing.

2.4 Existing proposals for integrating MANET routing with Mobile IP

The existing proposals for integrating MANET routing with Mobile IP can be separated into two groups: methods based on on-demand MANET routing protocols, and methods based on proactive MANET routing protocols. In this section, we review these existing proposals for integrating MANET routing with Mobile IP.

2.4.1 DSR with Mobile IP

A design for integrating the Dynamic Source Routing protocol (DSR) with Internet routing and Mobile IP is presented in [17], and its implementation is described in [18]. Mobile IP FAs are utilized to forward packets between a MANET and the Internet and to provide Mobile IP service for MNs in the MANET.

In the FA discovery procedure, this approach gives up the basic Mobile IP mechanism of propagating Agent Advertisements periodically. Instead, Agent Solicitations are utilized to detect the presence of an FA. An MN can know that it has entered a DSR network when it overhears DSR control packets such as Route Request or Route Reply. If the MN wants to participate in the MANET and to continue to receive Mobile IP service, then it will broadcast an Agent Solicitation piggybacked on a Route Request message. The Agent Solicitation is propagated over multiple hops through the MANET. When an FA receives the Solicitation, it unicasts an Agent Advertisement back to the MN to allow the MN's registration. Then, the MN registers itself with its HA through the FA.

In the Routing and Addressing procedure, the data transmission within a MANET is accomplished using the DSR protocol, while IP routing is used to handle data transfer between MNs in a MANET and nodes outside the MANET. DSR is both on-demand routing protocol and source routing protocol. IP routing on wired networks is proactive hop-by-hop routing. So, incorporating the two different routing mechanisms in the hybrid network architecture is a problem. Figure 2.2 shows a route discovery process when the destination of a packet is a node on the Internet. When node A needs to send packets to node D on the Internet, it initiates a route discovery procedure by propagating a Route Request. Node B and C add their address to the route list and repropagate the Route Request. Finally, the FA receives the Route Request and consults its route table. If the FA finds the entry to D in the route table, it sends a Proxy Reply including itself as second-to-last node in the route list to A. In the Proxy Reply, the reserved gateway interface index is used to differentiate this reply from normal Route Reply.

When node A originates a data packet to D and sends it to the FA by DSR routing, the FA can know that the destination is outside the MANET by examining the gateway interface index. So, the FA removes the source routing header from the packet and then transfers the packet to D by normal IP forwarding.

When a packet tunneled by HA is transferred to the FA, the FA transmits the packet to the destination in the MANET. If a source route cannot be found in the route cache, then the FA performs a route discovery and finally transfers the packet to the destination in the MANET by source routing.

Figure 2.2: Route discovery for node D on the Internet

2.4.2 MIPMANET

Ulf Jonsson and Fredrik Alriksson present a solution for connecting MANETs to the Internet, which is called MIPMANET (Mobile IP for Mobile Ad Hoc Network) [11]. The solution is based on AODV routing protocol [25] and Mobile IP. In this approach, an MN in a MANET gets access to the Internet and Mobile IP service using an FA CoA and reverse tunneling. FAs serve as access points to the Internet in order to keep track of

which MANET an MN is located in and to tunnel packets from MNs to destinations on the Internet. The AODV protocol is used to support routing between FAs and MNs within a MANET. MNs in a MANET use their Home address for all communication and to register with an FA.

In the FA discovery procedure, MIPMANET makes use of flooding Agent Advertisement messages to propagate the presence of an FA throughout a MANET. To reduce the cost of bandwidth and energy due to the Agent Advertisement flooding, the proposal uses a longer interval between consecutive advertisements. In Mobile IP, the minimum time between two consecutive Agent Advertisements is one second. The proposal suggests an interval time of five seconds. However, this tradeoff has an affect on Mobile IP. The FA discovery and movement detection process could be affected negatively since the information about an available FA is spread less frequently in the MANET.

An MN detects its arrival to a new MANET by receiving an Agent Advertisement indicating the availability of a new FA. Since there can be multiple hops between the FA and the MN, movement detection methods used in Mobile IP, such as Lazy Cell Switching (LCS) and Eager Cell Switching (ECS), are unsuitable. A new algorithm called MIPMANET Cell Switching (MMCS) is designed to determine when MNs in a MANET should register with a new FA. According to this algorithm, an MN should register with a new FA when it is at least two hops closer to this new FA than to its current FA, for two consecutive Agent Advertisements. The algorithm allows an MN to switch back and forth between two FAs in a controlled manner. It reduces the probability of temporary registration with an FA.

In the registration procedure, the main difference between MIPMANET and pure Mobile IP is that multiple hops are required for transferring the Registration Request and Reply messages between FAs and MNs. When an MN unicasts a Registration Request to an FA, it can initiate a route discovery procedure by sending RREQ and receiving RREP to find a current route to the FA. The FA registers the MN in its visitor list and forwards the Registration Request to the MN's HA. When a Registration Reply from HA is received, if the FA's route to the MN has timed out, then the FA can initiate a route discovery procedure to get a route to the MN.

In the routing and addressing procedures, the main difficulty in MIPMANET, compared with Mobile IP, is how to decide whether the destination of a packet is within the MANET or not. To overcome the gap between the proactive routing of Mobile IP and reactive routing of AODV, MIPMANET incorporates the default route concept into AODV. When an MN sends a data packet out, it first initiates a route discovery procedure to look for the destination in the MANET. If the destination is not found in the MANET, then the MN concludes that the destination is outside the MANET. Then, the MN encapsulates the packet to an FA with which it registers. The FA decapsulates the packet and forwards it to the destination on the Internet.

2.4.3 Global connectivity for IPv6 MANETs

In a MANET, MNs request setting up routes to the Internet and receiving Mobility support in IPv6 services. Ryuji Wakikawa et al. specifies a method for global connectivity for IPv6 MANET networks [31]. In their method, FAs are never needed but a gateway is utilized to build the connectivity between MANET nodes and Internet

nodes. MNs arriving in the MANET can acquire a global address from the gateway and communicate with its HA and CNs over the gateway. The process to accomplish this includes using a temporary address once an MN enters a MANET, which allows only local communication within the ad hoc cloud. Next the MN needs to discover a gateway and use its prefix to configure a globally routable IPv6 address. Then the MN does the normal Mobility support in IPv6 registration procedure and sets up routes to the Internet.

Whenever an MN enters into a MANET and needs to communicate over the Internet, the MN discovers a gateway to get the current global location information. The MN can use an arbitrary address that could be its home address or a site-local address by the address configuration to undertake the gateway discovery. The gateway discovery procedure can be done in the following two ways: the first one is by changing RREQ/RREP in on demand routing protocol, and the second one by changing NDP in IPv6 protocol.

In the first way, to support this method for global prefix information, route request and route reply messages of an on demand MANET routing protocol should be modified to carry global prefix information and the gateway's IPv6 addresses. In [31], researchers present an implementation of AODV6 operation with global connectivity for IPv6 MANET, where a modified RREQ and a modified RREP are used to attain global connectivity information proactively. When an MN requests global connectivity information, the node broadcasts a RREQ message to the global Internet gateway multicast address. Once a gateway receives the request, it must unicast a RREP message that includes both the global prefix and its IPv6 address back to the MN. Noticeably, the RREP must have a global prefix flag to clarify whether or not the route reply carries the gateway information and address.

The second way utilizes the IPv6 Neighbor Discovery Protocol (NDP) to obtain gateway prefix information, where route advertisement and route solicitation messages are modified. An obstacle to making use of NDP is that NDP only supports link-local scoped addresses. A multi-hop MANET needs global addresses. The proposal in [31] supports global scope instead of only link-local scope addresses to solve the problem. An MN can request a Router Advertisement to the gateways by sending a MANET Router Solicitation to the Internet gateway multicast address. The node uses an expanding ring search technique to broadcast the Router Solicitation message to the Internet gateway address using appropriate hop limit values. Those intermediate nodes, except for the gateways, do not process the Router Solicitation packet but forward it if the hop-limit allows this. If a gateway receives a Router Solicitation packet, it should reply with a MANET Router Advertisement message including its global prefix and its IPv6 address. Another problem solved by this technique is the need to set up the reverse route path for replying. A gateway generates a response to cause a search for the reverse route by AODV6.

Once the MN receives a route reply or a route advertisement, it uses the prefix from the gateway to configure a globally routable IPv6 address as its new CoA. The MN can send BU messages with the new CoA to its HA for registration. After finishing the registration with its HA, the MN can make global connectivity with nodes on the Internet.

In the routing and addressing procedure, the MN performs a route search algorithm for the destination. When the MN desires to send a data packet out, it first queries its route table to check if there is an unexpired route to the destination. If such a route is found, the MN knows the destination is inside the MANET, and it can send the packet to the

destination. If the only host route found is a default route (gateway), the MN initiates a route discovery process to search for a route to the gateway. Then the MN sends the packet to the destination by the gateway.

Once a route to the destination is known, the MN can send a packet using one of the following approaches. The first approach is called next hop routing. In this method, the MN sends a packet to the global destination address depending on the next hop routing of the other nodes. The second way is the use of a routing header. The MN uses a routing header in which the real destination address is inserted. The packet is tunneled to the gateway. Only the gateway can examine the routing header and route the packet to its final destination.

A comparison between the two approaches in [31] reveals that the Next hop routing approach has a smaller packet header size and requires less change on IPv6. The Routing Header approach can detect the use of incorrect routes.

2.4.4 RIP with Mobile IP

One of the earliest methods for facilitating Internet connectivity for MANETs is described in [14]. The method makes use of a modified version of the Routing Information Protocol (RIP) [8] to handle the connectivity within a MANET. The Mobile IP protocol is utilized to support the connection between MNs in a MANET and nodes on the Internet.

One source of the method is the work presented in [26], whose main idea is to treat each node as a route and to run a modified RIP. To accommodate the dynamic, self-starting behavior of MNs in MANET, three modifications are introduced in the RIP protocol.

First, each route table entry is assigned a sequence number, which originated with the destination node, to avoid the routing loop problem. Second, in order to dampen fluctuations in route table entries, each destination has an average settling time to delay the advertisement of those routes that may not have been settled. Third, once any topological change is noticed, route updates are propagated throughout the whole MANET.

The work in [14] extends the work of [26] to provide Internet access to a MANET. In this approach, an FA is used to support the data transfer between MNs in the MANET and CNs on the Internet. The software architecture of the method is shown in Figure 2.3. In this architecture, the Mobile IP protocol is executed by a *mipd* daemon, and an improved version of RIP is supported by a *routed* daemon. Both daemons need to manipulate the kernel route table. Hence, a route manager program, *rtmgrd*, is developed in the approach to coordinate the route manipulation operation requested by *mipd* and *routed*. *Rtmgrd* runs as a separate daemon starting before *routed* and *mipd* and terminating after them. *Routed* and *mipd* always send their route manipulation requests to *rtmgrd*. According to the source of the request and the history of route operations, *rtmgrd* decides on an appropriate action to be taken in the kernel's route table.

In the FA discovery procedure, the method utilizes the improved routing daemon *routed* to propagate Agent Advertisements to MNs multiple hops away from the FA. According to the Mobile IP protocol, *mipd* running on an FA broadcasts Agent Advertisement messages to declare the existence of the FA. On the FA, the *routed* process picks up the Agent Advertisements from the loopback interface and piggybacks it in the route table

entry for the FA. When the route table entry for the FA is propagated in the MANET, the Agent Advertisement can reach the *routed* process on each MN throughout the MANET. When an MN outside the communication range of the FA receives the Agent Advertisement through the *routed* daemon, the *routed* process relays the advertisement to the local *mipd* process. Upon reception of the Agent Advertisement, *mipd* on an MN can execute the registration to the FA. At the same time, the MN establishes the FA as its default router and adds a host route to the FA in the kernel's route table.

Figure 2.3: Software architecture of RIP with Mobile IP

In the routing and addressing procedure, the IP and the *mipd* daemon directly support data transfer to or from the Internet. When an MN in a MANET sends packets to a destination outside the MANET, the IP first searches its route table to find a default route. If the default route is the FA, the IP performs another look-up to find a route to the FA. When a packet directly received from a CN or tunneled by HA arrives at an FA, the FA delivers the packet to the current location of the destination by checking its visitor list.

2.4.5 MEWLANA-TD

In [6], researchers present Mobile Enriched Wireless Local Area Network Architecture (MEWLANA) that covers the table-driven routing type by proposing the MEWLANA-TD protocol. The MEWLANA-TD uses the DSDV protocol, a table-driven MANET routing protocol in which each node has an entry in their route table for every other node. In this method, FAs serve as access points for MNs in a MANET to connect to the Internet. The network architecture is shown in Figure 2.4.

Figure 2.4: Network architecture of MEWLANA-TD

In the FA discovery procedure of MEWLANA-TD, Agent Advertisement messages are duplicated by MNs and spread through the depth of the MANET. Agent Advertisement messages include CoA, source address and hop count. CoA identifies the new location of an MN for accessing the Internet. The source address is the address of the node sending the Agent Advertisement, which is varied in the spreading process. The hop count is increased each time an advertisement message is duplicated. The hop count gives an MN knowledge of the distance to the FA so that the MN can detect the nearest FA while it is

connected to multiple FA domains. Additionally, a spreading process of Agent Advertisement messages can be terminated by setting a maximum hop count. Each time an MN receives an Agent Advertisement, it sets the source address in the message as its default gateway and record its new CoA. Then, the MN increases the hop count and changes the source address using its address. After updating the Agent Advertisement message, the MN sends the message again to other nodes in the MANET. As a result, the next hop of the path from an MN to the FA is recorded as the default gateway entry in the route table of the MN. For example, in Figure 2.4, MN3 records MN2 as the next hop to the default gateway in its route table.

To reduce the overhead of flooding Agent Advertisement messages through a MANET, in the MEWLANA-TD proposal, dynamic beaconing algorithm is introduced in the FA discovery procedure. Since DSDV is a table-driven protocol, route tables can be utilized to monitor the changes of the network architecture of a MANET. MEWLANA-TD limits the advertisement message beaconing by checking the changes in its route table. These changes include node join or leave and route change. Agent Advertisements are initiated only when the route table changes.

In the Registration procedure of MEWLANA-TD, an MN sends a registration request to its HA via the FA. The FA records the MAC/IP address entry for the MN in a visiting nodes table according to the received registration request. For example, in Figure 2.4, FA records the MAC of MN1 and the IP address of MN3 for MN3. MN1 is viewed as the gateway to MN3 by the FA.

In the routing and addressing procedure of MEWLANA-TD, as each MN has entries to all other nodes and default gateways in the route table, the MN can decide if the

destination of a data packet is inside or outside the MANET by checking its route table. The tunneling procedure in MEWLANA-TD between HA and FA is the same as the basic Mobile IP.

2.4.6 CRC OLSR with Mobile IPv6

The Communication Research Centre Canada (CRC) presents a new approach to integrating WLANs and MANETs into the IPv6 based Internet. In the proposed network architecture (Figure 2.5), the Optimized Link State Routing (OLSR) protocol is employed for routing within the MANET. The transfer of information into and out of the MANET is facilitated through a MANET gateway located between the MANET and the Internet. Location management is achieved through Mobility support in IPv6. Moreover, a test-bed is constructed to demonstrate the viability of the proposed approach. The test results for performance benchmarking of this test-bed are presented in a technical report for INSC (Interoperable Networks for Secure Communications) [32].

Figure 2.5: WLANs & MANETs based Mobile network [15]

The MANET gateway is responsible for understanding the routing scheme of the Internet and the OLSR based routing protocol in the MANET. In the gateway discovery

procedure, a built-in OLSR message, Host Network Association (HNA) message, is exploited to propagate the address information of the gateway in the MANET. When an MN moves into a MANET, it first use its home address to join the MANET and listen to the HNA message. Upon receiving an HNA message, the MN can autoconfigure its CoA by combining the MAC address and network prefix contained in the HNA message. Then, the MN sends a BU message to the remote HA and CNs through the gateway.

In the routing and addressing procedure, the proactive routing feature of OLSR protocol improves data transfer between MNs in the MANET and CNs on the Internet. When an MN sends packets to a CN outside of the MANET, it can transfer the packets to the gateway by checking the default router entry in its route table. When packets from the Internet are received by the gateway, the gateway can forward the packets immediately to the MN since there is a route to the MN's CoA kept in the gateway's route table.

In addition, this approach introduces an automatic mode-detection and switching capability in each MN to facilitate handoffs between WLANs and MANETs. In the proposed network architecture, each MN uses the 802.11 adapter card to access the WLAN or the MANET. The IEEE 802.11 standard defines two basic modes of operation for wireless networks: the Ad-hoc mode and the Infrastructure mode. However, this standard does not define the handoff procedure between the two 802.11 operational modes. In [15], an algorithm is designed to support the handoff between the two distinct 802.11 operational modes. This algorithm includes mode-detection and mode-switching procedures.

2.5 Performance analysis of existing integration methods

In Section 2.4.2, several existing methods of integrating MANETs to the Internet, some of which are based on on-demand ad hoc routing protocols while others are based on table-driven routing protocols, are reviewed. Most of these proposals are implemented and performance measurements are taken in a simulated environment or on a real testbed. The performance analysis on the integration methods mainly focuses on two areas: protocol overhead and handoff latency. The protocol overhead includes “Ad hoc Routing Overhead” (AHRO) and “Mobile IP Overhead” (MIPO) [6]. The handoff latency is defined as the interruption time between the end of attachment to the old subnet and attachment to the new subnet.

Generally speaking, the methods based on table-driven routing protocols generate high AHRO because of routing information exchanged periodically among nodes in a MANET. The methods based on on-demand routing protocols reduce AHRO by initiating route discovery messages only when the route is needed. However, the on-demand feature in MIPMANET or DSR with a Mobile IP method results in longer handoff latency than those using table-driven integration methods. The MIPO in the existing proposals is the beacon flooding of Agent Advertisements or gateway route advertisements. To reduce the MIPO, some proposals introduce schemes to decrease the transmission frequency of Agent Advertisements, such as the dynamic beaconing algorithm in MEWLANA-TD.

Since these existing proposals have different performance features, we could not judge which is the “optimum” method. Different user expectations in different environments could require different methods. In [6], the application environment is classified based on

the MANET network size and intensity level of inside traffic. In an environment where the majority of traffic is inside the MANET, the transmission of Agent Advertisements need not be so frequent because only a minority of MNs requests the service of the FA. On the other hand, when the size of a MANET increases, the performance of table-driven methods degrades because of the overhead of the route table exchanges. For example, the MEWLANA-TD proposal is a table-driven routing protocol and uses a dynamic beaconing algorithm to limit the flooding of Agent Advertisements. Therefore, [6] indicates that the MEWLANA-TD proposal is efficient for those hybrid networks that have a small network size and high inside traffic.

In designing a method to support the integration of MANETs with the Internet, ways of reducing MIPO and AHRO need to be considered. On the other hand, handoff latency is also an important metric for providing an efficient Mobile IP service for MNs in MANETs. These factors are the motivation for our work. The design of our approach takes into consideration the effects of these factors.

Chapter 3 OLSR and HMIPv6 Review

In the previous chapter, we reviewed the basic MIPv6 protocol and different types of MANET routing protocols. In our approach to integrating MANETs into the Internet, Hierarchical Mobile IPv6 (HMIPv6) protocol is employed to support the mobility management across MANETs and WLANs; Optimized Link State Routing (OLSR) protocol is chosen as the internal routing protocol within a MANET. To interpret the motivation for employing HMIPv6 and OLSR as the basic mobility management schemes in our approach, we review HMIPv6 protocol and OLSR protocol in this chapter. As such, Section 3.1 and 3.2 review HMIPv6 protocol and OLSR protocol respectively.

3.1 HMIPv6

HMIPv6 as an extension of the basic MIPv6 is presented in [3]. The protocol proposes a hierarchical scheme that differentiates local (intra-site) mobility from global (inter-site) mobility in the Internet. Based on the hierarchical network architecture, HMIPv6 employs different network servers as mobility management agents to manage different levels of mobility of a mobile node. In sub section 3.1.1, we give the overview of HMIPv6. In sub section 3.1.2, we review the detail of regional mobility, local mobility and data transfer operations. In sub section 3.1.3, we compare the performance between the basic MIPv6 and HMIPv6.

3.1.1 HMIPv6 overview

In HMIPv6, an MN's CoA is split into two Care-of-Addresses:

- Regional CoA (RCoA), an address obtained by the MN from the visited domain
- Local CoA (LCoA), an on-link CoA configured on an MN's interface based on the prefix advertised by its default router

The two CoAs are used to handle global mobility and local mobility respectively. The RCoA identifies the domain where an MN is visiting. The LCoA identifies the MN's access point within the domain.

To build an RCoA and manage local mobility in HMIPv6, a new entity called a Mobile Anchor Point (MAP) is introduced. The MAP functionality can be installed on a domain MAP server. A logical tree structure in the domain with MAP as root is shown in Figure 3.1.

The existence of a domain MAP is advertised by access routers (ARs) as a new MAP option in the Router Advertisement message. An MN can finish the stateless autoconfiguration of its new RCoA by receiving a new MAP prefix. An MN registers its LCoA with the MAP and only registers its RCoA with its HA and CNs. So, when an MN moves within a domain, it does not need to re-register its local access point with its HA and CNs outside the domain. This means that, during local mobility, the MN's RCoA, known by its HA, remains unchanged so that global travel of the registration signal is avoided.

Figure 3.1: Hierarchical tree structure in HMIPv6

Such a hierarchical approach has at least two benefits. First, it reduces signaling latency; since signaling messages travel only up to the MAP for local handoff, therefore local mobility performance is improved. Second, it reduces signaling overhead; since mobility control messages are limited to a local scope, the signaling load on the whole Internet is reduced.

HMIPv6 has two modes based on the usage of RCoA. One is the basic mode: an MN forms its own RCoA on the MAP's subnet. Another is the extended mode: an MN is

configured with an RCoA that is assigned to one of the MAP's interfaces. The implementation of the basic mode is simpler than that of the extended mode.

Currently, most HMIPv6 implementations only support the basic mode. As a result, the basic mode of HMIPv6 is exploited in our approach.

3.1.2 HMIPv6 operation

In HMIPv6, an additional MAP discovery process is introduced. HMIPv6 considers local mobility and global mobility as two different things. In the following, the main operations of HMIPv6 will be described in more detail.

- **MAP discovery**

A domain MAP server announces its existence periodically using the MAP options in the Router Advertisement messages. The ARs are required to forward the MAP options in all Router Advertisement messages. A MAP option includes the distance vector from an MN in terms of the number of hops which is initialized to one, the MAP's global IP address and the MAP's subnet prefix. Upon reception of a Router Advertisement message with the MAP option, an AR copies the option and re-sends it after incrementing the distance field by one.

An MN can learn of the availability of the local domain MAP servers by receiving RAs with the MAP options.

- **Regional mobility**

When an MN moves from an old domain to a new domain, it will do movement detection, CoA configuration and registration. The process is shown in Figure 3.2. There are two steps:

Step 1: Movement Detection & CoA configuration

An MN can detect its arrival to a new domain and learn about the availability of a new MAP server through the MAP options in Router Advertisements (RAs). (See number 1 in Figure 3.2.)

After receiving RAs, the MN can configure its RCoA using the MAP prefix in RAs and its LCoA using the attaching subnet prefix in RAs.

Step 2: Registration

1. Registering LCoA with a MAP server (see number 2 in Figure 3.2)

The MN sends Binding Update (BU) messages that specify the binding between its RCoA and its LCoA to the domain MAP server. Upon reception of this BU, the MAP server performs admission control by checking the authentication field. If the request is accepted, the MAP server sends ACKs back to the MN.

2. Registering RCoA with its HA (see number 3 in Figure 3.2)

After accepting the ACKs from the domain MAP server, the MN sends BU messages that specify the binding between its home address and RCoA to its HA. The HA performs admission control by checking the Authentication field and returns ACK to the MN.

3. Registering RCoA or LCoA with CNs

The registration process with a CN involves two cases: when the CN is on the same link as the MN, and when the CN is on a different link. The MN registers its RCoA with CNs on the different link by sending BU messages that specify the binding between its home address with its RCoA (see number 3 in Figure 3.2). The MN registers its LCoA with CNs on the same link by sending BU messages that specify the binding between its home address and its LCoA (see number 4 in Figure 3.2).

Figure 3.2: Regional mobility in HMIPv6

- **Local mobility**

When an MN moves from an old subnet to a new subnet within the same domain, it performs movement detection, CoA configuration and registration. The transitions are performed locally. The process is shown in Figure 3.3. There are two steps:

Step 1: Movement Detection & CoA configuration (see number 1 in Figure 3.3)

An MN monitors the availability of the previous discovered domain MAP according to received RAs. While the MAP does not change, the mobility is considered local. Then, the MN configures its LCoA using subnet prefixes in the RAs. The MN's RCoA is kept unchanged.

Step 2: Registration

1. Registering LCoA with a MAP server (see number 2 in Figure 3.3)

The MN sends BU messages that specify the binding between its RCoA and its new LCoA to the domain MAP server.

2. Registering LCoA with CNs on the same link (see number 3 in Figure 3.3)

The MN sends BU messages that specify the binding between its home address and its new LCoA to those CNs on the same link.

3. Registering RCoA with CNs on the old link (see number 4 in Figure 3.3)

The MN sends BU messages that specify the binding between its home address and its current RCoA to those CNs on the old link.

Note that during the local mobility, no BU is sent on the Internet network outside the local domain.

Figure 3.3: Local mobility in HMIPv6

- **Data transfer**

In the data transfer process, a MAP server takes the role of a local HA. The MAP server can intercept and tunnel packets to MNs. The process is shown in Figure 3.4.

A CN on a different link sends packets to an MN using the MN's RCoA as the destination address. The packets are intercepted by a domain MAP server, which the MN registers with. Then the MAP server tunnels the intercepted packets and forwards them to the current LCoA of the MN (see number 1 and 2 in Figure 3.4). A CN on the same link sends packets to an MN using the MN's LCoA as the destination address. Packets are directly delivered to the current LCoA of the MN (see number 3 in Figure 3.4).

An MN sends packets to CNs outside the domain by using a normal IPv6 routing mechanism. The MN's LCoA or RCoA can be chosen as the source address. The home address of the MN is included in the routing header of the IPv6 packet.

Figure 3.4: Data transfer in HMIPv6

3.1.3 Performance comparison of basic Mobility support in IPv6 and HMIPv6

To explain the reason why the HMIPv6 protocol is employed to support the handoff between MANETs and WLANs in our approach, we compare the performance of basic MIPv6 and HMIPv6. When comparing the performance of different mobility management mechanisms, three factors are particularly important [16]. The first is the handoff performance, i.e. how fast is the handoff performed. The second is the routing performance, i.e. what is the extra latency for data transfer introduced by each mechanism. The third is the protocol overhead, i.e. how many control messages are generated for mobility management. In the following, we compare the performance of MIPv6 and HMIPv6 according to the effect of the above three factors.

- **Handoff performance**

We do the performance comparison between the two mobility management schemes. We introduce the scenarios and the parameters chosen for performing the analysis. In Figure 3.5 we can picture the network architecture used for the performance analysis.

There is an MN initially attached in AR1. The MN receives data packets from the CN. Since the purpose of HMIPv6 is to reduce the amount of control messages to CNs and HA and in some cases improve the handoff performance, the introduction of a hierarchy is relevant only if the MAP is located between the MN location and the HA and CN locations. Therefore, in the studied scenarios, the MAP is located on a route between the MN location and the HA and CN locations.

Figure 3.5: Studied scenarios

The study will cover two typical scenarios: when the MN does a handoff from AR1 to AR2, and when the MN does a handoff from AR2 to AR3. To simplify the parameters, the local wired link delays are assumed to be the same and referred to as ld_{lo} ; the wired link delay between HA and the domain router is referred to as ld_{HA} ; the wired link delay between CN and the domain router is referred to as ld_{CN} ; and the wireless link delay is referred to as wd . On the other hand, we assume that processing delays are negligible.

Because the smooth handoff (a previous AR can forward packets from CNs to a new AR) is allowed in basic MIPv6, the packets are sent by the CN either to the HA or to the previous AR (PAR) depending on whether the entry in the binding cache for the old CoA has expired or not. Therefore, the handoff latency using basic MIPv6 is directly proportional to the round-trip time necessary for a BU to reach either the HA, the CN or the PAR. In using HMIPv6, the handoff case is divided into two scenarios: local handoff and global handoff. The local handoff latency is proportional to the round-trip time

between the MN and the MAP or the PAR. The global handoff latency is proportional to the round-trip time between the MN and the HA, the CN or the PAR.

Considering the time for a MN to send out a BU and receive an ACK, the handoff latency under the two scenarios is shown in the following:

1) MN moves from AR1→AR2

For basic MIPv6, the handoff latency is:

$$L_{mip} = 2 \times (wd + ld_{lo} + \min(ld_{HA}, ld_{CN}))$$

if the binding entry for the MN's old CoA has expired.

$$L_{mip} = 2 \times (wd + ld_{lo} + ld_{lo})$$

if the binding entry for the MN's old CoA has not expired.

For HMIPv6, the handoff latency is:

$$L_{hmip} = 2 \times (wd + ld_{lo})$$

If $ld \gg Wd$, i.e. the HA and CN are located in the remote network far away from the foreign network, then L_{hmip} is much smaller than L_{mip} under this scenario.

2) MN moves from AR2→AR3

In the scenario, the behavior of HMIPv6 is similar to that of basic MIPv6, so the two schemes have quite similar handoff latencies.

Hence, the HMIPv6 achieves better handoff latency than the basic MIPv6 when the mobility is local.

- **Routing performance**

With basic MIPv6, the routing is optimal. Packets follow the path from the CNs to the MN directly except for the first packet which goes through the HA. With HMIPv6, an extra indirection through the MAP is required. However, the cost of this indirection is small especially if the MAP is close to the border router as suggested in the HMIPv6 protocol [3].

- **Protocol overhead**

We compare the protocol overhead and extra signaling load on the Internet. Table 3.1 shows the control messages sent and received by an MN during each local handoff or each regional handoff.

		HA	CN	MAP
HMIPv6	Local Handoff			1 BU, 1 ACK
	Regional Handoff	1 BU, 1 ACK	1 BU, 1 ACK	1 BU, 1 ACK
MIPv6	Handoff	1 BU, 1 ACK	1 BU, 1 ACK	

Table 3.1: Control messages sent and received by an MN in a handoff

According to Table 3.1, we calculate the amount of control messages exchanged during each handoff with basic MIPv6 or HMIPv6. We refer the number of CNs as N_{CN} .

During each handoff, the number of control messages generated by basic MIPv6 is $2*(1+N_{CN})$ since one BU and one ACK are exchanged between the MN and the HA as well as each CN. The number of control messages generated by HMIPv6 is two during each local handoff; the number is $2*(1+1+N_{CN})$ during each regional handoff since the CN exchanges control messages with the MAP, the HA and each CN.

We observe that the protocol overhead of HMIPv6 is much smaller than the one of basic MIPv6 if the MN only moves within a domain, while the protocol overhead of HMIPv6 consists of two more messages to the basic MIPv6 if the MN moves inter-domain. However, if we evaluate the signaling load of the two schemes, we find that HMIPv6 results in less signaling load on the Internet backbone than MIPv6 if the main part of the handoff is the local handoff. Based on the consideration that local resources are not the most critical [3], we can assert that HMIPv6 has performs better, in terms of signaling load, than does basic MIPv6.

Based on the above performance comparison, we conclude that HMIPv6 improves on basic MIPv6 in supporting the seamless handoff for MNs traveling frequently within a domain. Therefore, we choose HMIPv6 protocol to achieve mobility management across MANETs and WLANs in our approach.

3.2 OLSR protocol

We review OLSR protocol in this section. In sub section 3.2.1, we give the overview of HMIPv6. In sub section 3.2.2, we review the details of the main operations of OLSR. In sub section 3.2.3, we introduce OLSR networking support. Finally, we interpret the salient features of OLSR for facilitating the integration of MANETs with the Internet in sub section 3.2.4.

3.2.1 OLSR overview

OLSR is a table-driven routing protocol standing for Optimized Link State Routing in MANETs [9]. Compared with on-demand routing protocols, OLSR has the advantage of

having the routes available immediately as needed. On the other hand, to reduce the bandwidth cost in exchanging frequent periodic control messages in the pure link state protocol, OLSR introduces a new technique called the Multipoint Relays (MPRs). Each node selects a set of nodes from its one-hop neighbors, which retransmits its broadcast packets. These selected neighbors consist of an MPR set of the node. To let a node decide if it should retransmit packets coming from other nodes, each node maintains information about those neighbors that select it as an MPR, which are called the MPR Selectors of the node. Based on the idea of MPRs, OLSR protocol achieves an optimization of the pure link state routing protocol as a result of two features. First, it reduces the size of control messages since it only declares the link with its MPR selectors instead of all links with its neighbors. Second, it minimizes flooding of the control messages as it only uses its MPRs to retransmit the broadcast messages in a MANET.

Each node selects its MPRs from its one-hop neighbors with a bi-directional link. The selection of an MPR set of a node follows a process that ensures the set covers all nodes that are two hops away from the node. The MPR nodes are selected as intermediate nodes in the routing path. To implement this scheme, each node periodically broadcasts the information about its MPR Selectors. Upon receipt of this MPR Selectors' information, each node calculates and updates its route table.

3.2.2 OLSR operation

In OLSR protocol, there are four main steps to creating a route table: neighbor sensing, MPR selection, MPR information declaration and route table calculation.

During the neighbor sensing, each node periodically broadcasts the HELLO message containing information about its one-hop neighbors and their link status. The HELLO messages are received by all the one-hop neighbors, but they are not retransmitted to further nodes. Upon receipt of the HELLO messages, each node can update its knowledge of its one-hop neighbors and two-hop neighbors. This information about one-hop and two-hop neighbors is recorded in a neighbor table. On the basis of this information, each node performs the selection of its MPR set.

During the MPR selection, each node independently selects its MPR set according to the MPR selection scheme. As a result, all of the two-hop neighbors of each node are contained in the union of the neighbor sets of its MPRs. Then, each node declares its MPRs in the subsequent HELLO messages. From the HELLO messages, which contain the MPRs, each node can inform its MPR Selectors and construct its MPR Selector table. During the MPR information declaration, each node broadcasts specific control messages called Topology Control (TC) messages to declare its MPR Selector set. The TC messages are forwarded through MPR nodes and transmitted to all nodes in the MANET. According to the MPR selectors and the information in TC messages, a node maintains a network topology table to record the MPRs of other nodes. The topology table is a base of calculating the route table.

During the route table calculation, a node calculates the route table based on the information contained in the neighbor table and the topology table. To find a path from a source (S) to a destination (D), an intermediate node (I_1) one hop away to D has to be found and the connection pair [I_1 , D] obtained; then, a node (I_2) one hop away to I_1 has to be found and the connection pair [I_2 , I_1] obtained; and so forth, until a node I_n is found

in the MPR sets of S . Based on this process, the route table is built by tracking the connection pairs included in the topology entries in a topology table. To attain the optimal routing path, the connection pairs on the minimal path are selected.

3.2.3 OLSR networking support

In OLSR draft version 5 [9], associated networks and host support are added. A node may act as a gateway between the MANET and a subnet not running OLSR. In order to inject the routing information describing the associated subnets into a MANET, a gateway periodically broadcasts its existence and its associated subnet address in the MANET using Network Association (HNA) messages. The HNA messages indicate which network is reachable through this gateway. The HNA messages are propagated in an optimal way so that only MPR nodes retransmit the broadcast messages.

Each node in the MANET maintains information concerning which node acts as a gateway and its associated network address. When a packet is destined for an address in a subnet indicated by a gateway and there is not a host entry in the route table, a MANET node will forward it according to the gateway entry in its route table.

3.2.4 Salient features of OLSR for facilitating the integration of MANETs with the Internet

To achieve the integration of a MANET with the Internet based IPv6, three main tasks need to be accomplished: gateway discovery, address autoconfiguration, and routing and addressing in a heterogeneous environment.

The first step is the gateway discovery. To achieve the hybrid network connection, the installation of gateways that understand the protocols of MANET and the IP suite is needed. From the point of view of MANET, these gateways act as ARs to the Internet. Upon initialization, an MN should discover the existence of gateways and then select one gateway as an AR to the Internet. The basic IP-based mechanism for access router discovery only supports those systems with one-hop connection between MNs and access routers. The MANET in the multi-hop environment makes the discovery mechanism more complex.

The second step is address autoconfiguration. In order to be able to communicate with Internet hosts, each MN must configure an IP address with the prefix of a gateway. With this address, the MN can send packets to and receive packets from hosts in the Internet.

The third step is routing and addressing in the heterogeneous environment. In a heterogeneous environment, when an MN arriving in a MANET sends packets to a host on the Internet, the packets are first transferred to a MANET gateway by MANET routing and then sent to the destination by IPv6 routing; when a host on the Internet sends packets to an MN arriving in a MANET, the packets are first transferred to a MANET gateway by IPv6 routing and then sent to the MN by MANET routing.

Some salient features of OLSR protocol provide benefits for the aforementioned steps. In the first step, the built-in HNA transfer mechanism in OLSR can be exploited to propagate the existence of gateways. The propagation mechanism provides an optimized route advertisement functionality for the multi-hop environment. In the third step, since OLSR is a table-driven routing protocol and it uses the same route table as IPv6 protocol, an MN can know if a destination is in the MANET or in the Internet by checking its route

table. No extra work is introduced to the GW for supporting the data exchange between an MN arriving in a MANET and a host on the Internet. This feature facilitates the routing and addressing process across MANETs and the Internet.

Compared with other MANET routing protocols, these salient features of OLSR can facilitate the integration of MANETs and the Internet. As a result, we choose OLSR as a routing protocol within MANET in our approach.

Chapter 4 Design of an OLSR with HMIPv6 Network

In Chapter 2, we investigated several strategies for supporting the integration of MANETs with the Internet. Here, we present a new scheme to integrate MANETs into the Internet and support mobility across WLANs and MANETs that are connected to the Internet. In this scheme, OLSR protocol is chosen to handle the internal mobility of mobile hosts within the MANET; HMIPv6 protocol is employed to support mobility management across MANETs and WLANs within a single domain or between different domains. Similar to the behavior of basic MIPv6, HMIPv6 is designed to support mobility where an MN is one hop away from the router. So, in our scheme, the challenge is to accommodate MANETs in such a way that an MN, which may be multiple hops away from a router, can be accessed from anywhere on the Internet. To overcome this challenge, the OLSR protocol and ICMPv6 are extended and modified in our scheme to satisfy the new functionality.

The chapter is organized in the following manner. Section 4.1 describes the architecture of an OLSR with HMIPv6 network. In Section 4.2 and 4.3, two important design decisions in our scheme are discussed, which are CoA autoconfiguration technique and join MANET mechanism. In Section 4.4, 4.5 and 4.6, the operations of three main processes including regional mobility, local mobility and data transfer are described respectively. Section 4.7 shows message flows in the main processes using the message sequence diagrams.

4.1 Network architecture

As illustrated in Figure 4.1, the proposed network consists of WLANs and MANETs connected to the Internet. The OLSR protocol is employed for routing within the MANET. Mobility management of an MN moving across WLANs and MANETs is achieved through HMIPv6.

In the hierarchical network architecture, a MANET is viewed as a basic unit of mobility management. MANETs and WLANs may belong to the same domain and use the same domain MAP server. MANETs and WLANs may belong to different domains and are managed by different domain MAP servers. The MANET gateway (GW), as a bridge connecting a MANET to the Internet, is responsible for transferring messages into or out of the MANET. The MANET GW is also used to propagate location information during an MN's movement.

An MN traveling in the proposed network has three IP address:

- Home address, to correspond to its home subnet and work as an identifier
- RCoA, to identify the domain region
- LCoA, to specify its location in the domain

4.2 CoA autoconfiguration technique

When an MN moves into a new domain, it should autoconfigure its new RCoA according to the MAP prefix. During a movement between WLANs and MANETs within the same domain, an MN keeps its RCoA. When the MN moves into a new basic unit area (a

Figure 4.1: Architecture of an OLSR with HMIPv6 network

WLAN or a MANET), it should autoconfigure its new LCoA according to the subnet's prefix.

In WLANs, since the MN is only one hop away from the AR, it can receive the RA with MAP option and the subnet prefix directly. The autoconfiguration of the MN's two kinds of CoA can be achieved using HMIPv6 scheme.

In MANETs, the MANET GW is used to receive and propagate the MAP option from the domain MAP. On the other hand, the GW should broadcast its existing and network prefix periodically to MNs arriving in the MANET. However, MNs that are more than

one hop away from the GW cannot receive the general RA message which can only be sent by link-scope multicast address. So, a new technique needs to be defined that allows a multi-hop distanced MN to receive the MAP option and GW address. Upon receipt of this, the MN can autoconfigure its RCoA and LCoA.

Two alternative ways are considered and compared in this design. One way is to modify the behavior of the RA message to support the multi-hop transfer. Another way is to use a new type of message called Auto Configuration Message (ACM) to achieve the transfer of MAP options and GW address in MANET. In the following sub sections, we interpret these two alternatives and give our design decision about CoA autoconfiguration.

4.2.1 RA approach

An original scheme in HMIPv6 to propagate MAP options and subnet prefixes through RAs can be modified to match MANET network architecture. Since the existing ICMPv6 RAs are designed for location detection in LANs where nodes are within the propagation scope of RAs, RAs will not be forwarded. Therefore, those hosts that are more than one hop away from the MANET GW will not receive the RAs. A significant modification to the IPv6 is required to accommodate the forwarding of RA messages on nodes in MANET.

The approach includes the following main functions: GW receives MAP option and piggybacks it in ICMPv6 RA; GW propagates RA with MAP option and network prefix using IP layer multicast address; intermediate nodes forward the RA; MN receives the RA, and extracts the MAP option and network prefix to configure its RCoA and LCoA.

This approach has the drawback of generating a high traffic load because of the ICMPv6 RAs flooding throughout a MANET.

4.2.2 ACM approach and design decision

In our network architecture shown in Figure 4.1, the selected OLSR protocol includes some optimal features to achieve efficient message transfer within a MANET. In OLSR draft version 5 [9], a GW periodically broadcasts its associated host and subnet in the MANET by using the Host and Association (HNA) message. This message is used to inject routing information into the MANET of those associated hosts/subnets that are not running MANET routing protocol. HNA messages are optimally broadcast to a selected set of nodes called MPRs instead of to all MANET nodes.

By referring to the optimal feature of HNA message propagation, we design an optimal mechanism to handle MAP advertisement and MN's CoA auto configuration issues. A new message type, the Auto Configuration Message, is added to OLSR. A GW periodically broadcasts its existence and MAP option as well as network prefix by propagating the ACM messages. The ACM messages are propagated through MPR in an optimal way. An MN arriving in the MANET uses the ACM message to configure its RCoA and LCoA instead of using the ICMPv6 RA message.

As with the RA approach, the ACM technique also involves some modifications to IPv6 in order to do CoA autoconfiguration using the ACM messages. However, the original RA message behavior is still kept. On the other hand, this approach has an advantage: it can make use of the OLSR optimal feature to reduce traffic overhead. Therefore, we use the ACM way to support MN's CoA autoconfiguration technique in our approach.

During the MAP discovery process, the domain MAP server broadcasts its existence periodically through MAP options. The MANET GW can receive a MAP option and convert it into an ACM message. The ACM message is broadcast in an optimal way

within the MANET. An MN in the MANET gets the domain MAP information through the received ACM message.

To implement the functionality, the additional GW operations include receiving MAP options, and constructing and propagating ACM messages. The additional MPR operations include receiving and processing ACM messages, creating a GW entry in the route table, and forwarding ACM messages to other nodes in MANET. The additional MN operations include receiving and processing ACM messages, creating a GW entry in the route table, and CoA autoconfiguration.

When an MN arriving in a MANET receives an ACM message from a GW and learns of the availability of a new MAP, the MN detects its regional movement. So, the MN autoconfigures its new RCoA using the MAP option in the ACM message. The RCoA autoconfiguration is processed in IPv6. In Chapter 5, we will explain the implementation of this process.

When an MN receives an ACM message from a GW and detects the availability of a new network prefix, the MN detects its local movement. So, the MN autoconfigures its new LCoA using the network prefix in the ACM message. The LCoA autoconfiguration is processed in IPv6. In Chapter 5, we will explain the implementation of this process.

4.3 Join MANET mechanism

After arriving in a MANET, an MN receives the ACM message from the MANET GW and configures its RCoA and LCoA. The MN has three types of address: Home address, LCoA and RCoA. So, when the MN sends new Hello messages to its neighbors, it can use its Home address, RCoA or LCoA as its identifier to join the MANET. The address

is used by other nodes to construct the topology table within the MANET. As a result, we have three choices in designing the mechanism to join the MANET, using either the MN's Home address, RCoA or LCoA.

In the following sub sections, we analyze the advantages and disadvantages of each choice and make our design decision. To differentiate CNs in different locations, we refer to a CN located outside the MANET as an external CN and to a CN located inside the MANET as an internal CN.

4.3.1 Using home address

If an MN uses its Home address to join a MANET, other nodes within the MANET will recognize it and connect to it by its Home address. Since a MANET GW is a bridge connecting the MANET with the Internet, an incoming packet from an external CN to the MN will be sent to a MANET GW first by HMIPv6 routing mechanism, then the packet will be forwarded to the current location of the MN by the OLSR routing mechanism. The destination of the packet arriving the GW is the MN's LCoA. However, other MANET nodes only know the MN's Home address, so the GW has to translate the MN's LCoA to the MN's Home address and then transfer the packet to the MN's current location through MPR nodes. Therefore, all packets to the MN have to be intercepted and translated by the GW. The additional operation process not only introduces more modification work in OLSR and IPv6 on the GW, but also affects the routing performance for data packets from an external CN to an MN arriving in the MANET. Furthermore, with the increase in the number of MNs arriving in a MANET and in the number of external CNs, the load on the GW increases substantially. The GW becomes a

bottleneck of the communication between MNs in a MANET and other external CNs on the Internet.

To conclude, this choice has the following advantages and disadvantages. The advantage is that internal CNs can send packets directly to MNs arriving in the MANET through OLSR routing. The disadvantage is the extra load that is added on a MANET GW, which could create a bottleneck in communications between MNs and external CNs.

4.3.2 Using RCoA

If an MN uses its RCoA to join a MANET, all nodes within the MANET recognize and connect it by its RCoA. According to HMIPv6 mechanism, a CN only knows the binding between an MN's Home address and the MN's RCoA. So, after receiving the MN's BUs with the MN's RCoA, internal CNs can send packets to the MN's current location directly through OLSR routing. However, similar to that mechanism using Home address to join MANET, all packets from external CNs have the MN's LCoA as destination address when they arrive at the GW. The GW has to intercept these packets and translate their destination address for the MN's RCoA. Then, the packets are transferred to the MN in MANET.

So, the mechanism has the following advantages and disadvantages. The advantage is that internal CNs can send packets directly to MNs arriving in the MANET through OLSR routing. The disadvantage is the extra load that is added on a MANET GW, which could create a bottleneck in communications between MNs and external CNs.

4.3.3 Using LCoA

If an MN uses its LCoA to join a MANET, all nodes within the MANET recognize and connect it by its LCoA. In our OLSR with HMIPv6 approach, an MN arriving in a MANET configures its LCoA using a GW's IPv6 address prefix and its MAC address. The packets from external CNs are sent to the GW and forwarded directly to the MN's current location by IPv6 data forward scheme. That means no extra modification work is needed on the GW. At the same time, the GW will not risk becoming a bottleneck for the connection between MNs in MANET and external CNs on the Internet.

However, the mechanism could affect the routing performance of data transfer from internal CNs to MNs in MANET. According to HMIPv6 mechanism, an MN's LCoA is transparent to its CNs. So, when a CN maintains connection with an MN arriving in the same MANET, it first sends the packets to the local domain MAP. Then the packets are encapsulated and forwarded to the MANET GW. Finally, the packets are transferred to the current location of the MN in the MANET.

Therefore, the mechanism has the following advantages and disadvantages. The advantage is that no extra load is added on a MANET GW for data transfer from external CNs to MNs arriving in the MANET, so the bottleneck problem in the aforementioned two ways does not appear. The disadvantage is that an extra indirection through the MAP and the MANET GW is required for data transfer from internal CNs to the MN. However, this disadvantage can be overcome as the discussion in Section 4.3.4 shows.

4.3.4 Design decision of joining MANET mechanism

To avoid the MANET GW becoming a bottleneck for the connection between an MN arriving a MANET and external CNs on the Internet, we think the mechanism using

LCoA to join MANET is a better scheme on routing performance compared with the other two mechanisms. On the other hand, we prefer to avoid doing the IPv6 modification work on a MANET GW that could be avoided in using these other mechanisms.

Therefore, to improve the routing performance for data transfer from internal CNs to an MN within the same MANET, we design a new registration scheme on MN using its LCoA to join a MANET. The main idea in the scheme is to differentiate internal CNs from external CNs and employ different registration operations to the two different types of CNs. An MN can register its RCoA with those external CNs by sending BUs with its Home address and its RCoA to those external CNs. Unlike external CNs, the MN can register its LCoA with those internal CNs, i.e. send BUs binding its Home address and its LCoA to those internal CNs. To implement this new registration scheme, the HMIPv6 registration process needs to be extended. An internal CN list needs to be built on the MN. The information of internal CNs can be obtained from the topology table in OLSR. So, communication between OLSR and HMIPv6 is necessary.

Based on the above analysis, we consider that the mechanism to join MANET using LCoA is the most effective approach.

4.4 New message structure

The OLSR with HMIPv6 scheme introduces two new message types to OLSR protocol and ICMP protocol: Auto Configuration Message (ACM), and local RA message.

4.4.1 ACM message

As described in Section 4.2.2, RA messages containing subnet prefix information and MAP information can only be propagated in one hop scope. In a MANET where MNs may be more than one hop away from the GW, the role of the RA is replaced by an ACM, which is a new message type added in OLSR protocol. The ACM message contains the MAP option and MANET GW network address as well as the two address' lifetimes.

The format of the ACM message is shown in Figure 4.2.

Figure 4.2: ACM message data structure

4.4.2 Local RA message

In original IPv6 and HMIPv6 protocol, the MAP option and subnet prefix are propagated in RA messages. When an MN receives an RA, it will do address auto-configuration work and other mobility management work by HMIPv6 functions. However, in our scheme, the address configuration information is transferred in ACM messages through an OLSR propagation scheme. When an MN receives an ACM message by OLSR routing, it has to transfer the address configuration information to IPv6 and HMIPv6. To achieve the transfer process, a new ICMP6 message is introduced in our scheme, which is

called the local RA message. The local RA message is used to carry the MAP option and GW prefix information from an application process to the Linux kernel. The local RA message is viewed as an extension of the normal RA message. The MAP information is put in the MAP option of the ICMP6 message, while the GW address information is put in the prefix option. On the other hand, to implement the handover between infrastructure networks and MANETs, it is necessary for an MN to differentiate between a local RA received from the local interface and a normal RA received from the link layer. To address this need, a new flag called an ACM flag is introduced in ICMP6 header to identify a local RA. The flag makes use of a reserved bit in icmp6hdr, which is defined in ICMP6. When a local RA is constructed, the ACM flag is set. The format of the local RA message is shown in Figure 4.3.

Figure 4.3: Local RA message data structure

4.5 Regional mobility

When an MN arrives in a MANET, it detects local movement or global movement by receiving and processing ACM messages. If the MAP option in an ACM message indicates the availability of a new MAP, the MN learns that it has entered a new domain and would handle regional mobility. The process is shown in Figure 4.4.

Figure 4.4: Regional mobility in OLSR with HMIPv6

After configuring its RCoA and LCoA, the MN finishes the following process:

Step 1: Exchange HELLO message carrying its LCoA with other MANET nodes in order to join the new MANET (see number 1 in Figure 4.4)

Step 2: Register LCoA with the MAP server (see number 2 in Figure 4.4)

MN sends BU messages that specify the binding between its RCoA and its LCoA to the domain MAP server. BU messages are transmitted to the MAP server through the MANET GW.

Step3: Register RCoA with HA and external CNs (see number 3 in Figure 4.4)

After accepting ACK from the MAP server, the MN sends BU messages that specify the binding between its home address and RCoA to its HA and CNs outside the domain. BU messages are transmitted to external Internet through the MANET GW.

4.6 Local mobility

When an MN arrives in a MANET, it detects local movement or global movement mode by receiving and processing ACM messages. If the MAP option in the ACM message indicates the availability of the same MAP but a new network prefix, the MN learns that it is still moving within the same domain and needs to handle local mobility. The process is shown in Figure 4.5.

After configuring its new LCoA, the MN finishes the following process:

Step 1: Exchange HELLO message carrying its LCoA with other MANET nodes in order to join the new MANET (see number 1 in Figure 4.5)

Step 2: Register new LCoA with the MAP server (see number 2 in Figure 4.5)

MN sends BU messages that specify the binding between its RCoA and LCoA to the domain MAP server. BU messages are transmitted to the MAP through the MANET GW.

For HA and external CNs the RCoA has not changed. No control message is transmitted on the Internet for the local mobility.

Figure 4.5: Local mobility in OLSR with HMIPv6

4.7 Data transfer

In the data transfer process, the MAP server serves as a local HA and can intercept and tunnel packets to MANET GW. Internal routing within the MANET is achieved by OLSR routing mechanism. The process is shown in Figure 4.6.

An external CN sends packets to an MN by using the MN's RCoA as the destination. A domain MAP server intercepts the packets (see number 1 in Figure 4.6). Then the MAP server tunnels the intercepted packets and forwards them to the GW of the MANET

where the MN is located (see number 2 in Figure 4.6). Within the MANET, the packets are transferred to the current LCoA of the MN through OLSR routing (see number 3 in Figure 4.6).

Figure 4.6: Data transfer in OLSR with HMIPv6

4.8 Message sequence diagram

In this subsection, we use message sequence diagrams to show message flows between different components in the handoff process and data transfer process.

4.8.1 Message sequence diagram in handoff process

In MN's handoff process (local handoff process or regional handoff process), nodes outside the MANET, including MAP, HA, and CN, and nodes within MANET, including

MANET GW, MPR and MN, could be involved. Figure 4.7 shows the control message flows among different components when an MN moves into a MANET. When the MN does a local handoff, no control message is sent to or received from HA or CN.

4.8.2 Message sequence diagram in data transfer process

In the data transfer process, MAP performs the role of local HA and can intercept and tunnel packets to MANET GW. Internal routing in the MANET is achieved by OLSR. A CN could be located in the MANET or in another network connecting to the Internet. An MN arriving in a MANET can send data packets to a CN within the same MANET directly through OLSR routing; it can also send data packets to a CN in another network through the MANET GW. However, based on HMIPv6 routing mechanism, a CN always sends data packets to an MN's RCoA. The packets are intercepted by the MAP of the domain where the MN is visiting. Then the MAP tunnels the intercepted packets to the GW of the MANET where the MN is arriving. Within the MANET, the packet is transferred to the current LCoA of the MN through OLSR routing. Figure 4.8 shows the data flows among CN, MN and MAP, as well as GW and MPR.

Figure 4.7: Message diagram in handoff process

Figure 4.8: Message diagram in data transfer process

Chapter 5 OLSR with HMIPv6 Implementation

In Chapter 4, we present a new scheme for integrating MANET with the Internet. In this scheme, the OLSR protocol is employed as the MANET routing protocol while the HMIPv6 protocol is employed as a mobility management protocol. To achieve this scheme, the Monash's HMIPv6 implementation and CRC OLSR with IPv6 (CRC OLSR6) implementation are chosen to support our OLSR with HMIPv6 implementation. Since the Monash's HMIPv6 implementation exists in the Linux kernel space while the CRC OLSR6 implementation is in user space, our implementation has the following features. The implementation is done both in kernel space and in user space. The IPv6 implementation and ICMP6 implementation in Linux kernel space are modified and extended to satisfy the needs of this integration. Communication between kernel space and user space are built into our implementation. In this chapter, we discuss the salient features and challenges in the implementation and then introduce the implementation of MANET GW and MN.

The chapter is organized in the following manner. Section 5.1 introduces the implementation environment. In Section 5.2, the whole architecture of the implementation is provided. Section 5.3 describes the communication mechanism between user space and Linux kernel space in the implementation. Section 5.4 talks about the inter-process communication (IPC) mechanism in the user space of the implementation. In Sections 5.5 and 5.6, the details of MN implementation and GW implementation are described.

5.1 Implementation environment

We exploit the OLSR protocol and HMIPv6 protocol to implement the integration of MANETs with the Internet. So, our OLSR with HMIPv6 implementation is built above the implementation of HMIPv6 and OLSR.

The implementation of HMIPv6 is usually approached as an extension of an existing implementation of basic MIPv6. The three types of entities involved in HMIPv6 are MAP, AR and MN. The MAP's functions include propagating MAP options using RA messages, receiving BU messages and maintaining a binding cache, and receiving and tunneling packets to MNs. The AR's functions are to receive, process and forward MAP options in RA messages. The MN's functions include receiving and processing RA messages, configuring RCoA and LCoA, registering LCoA with a MAP, and registering RCoA with its HA and CNs.

Until now, there have been three available HMIPv6 implementations: the INRIA HMIPv6 [2], TU-Berlin HMIPv6 [20] and Monash HMIPv6 [4]. The INRIA HMIPv6 runs on FreeBSD 3.4. The TU-Berlin HMIPv6 runs on Linux v2.4.7 and is based on HUT's MIPL 0.9 and Lars Fennerberg's RADVD 0.7.1. The Monash HMIPv6 runs on Linux v2.4.18-3 and is based on HUT's MIPL 0.9.3 and Lars Fennerberg's RADVD 0.7.1.

We decided to build our implementation on Linux because of its current popularity. In comparing the two HMIPv6 implementations on Linux, we found that both of them support the basic mode of HMIPv6, but the Monash HMIPv6 is more complete than the TU-Berlin HMIPv6. Therefore, we choose Monash's HMIPv6 as a HMIPv6 component of our OLSR with HMIPv6 implementation.

Since our implementation is built on IPv6 packet transfer, the OLSR implementation must support IPv6. The Communication Research Center of Canada (CRC) has an OLSR on IPv6 implementation (CRC OLSR6) [15] which completely achieves IPv6 packet transfer in MANETs through the OLSR protocol. We choose the CRC OLSR6 as an OLSR component in our implementation.

For the sake of consistency with the Monash HMIPv6, our OLSR with HMIPv6 implementation is also based on Linux v2.4.18-3.

5.2 Architecture of OLSR with HMIPv6 implementation

In order to achieve the OLSR with HMIPv6 scheme presented in Chapter 4, we modify and extend the basic CRC OLSR6, Lars Fennerberg's RADVD 0.7.1 and Linux kernel to support GW functionality and MN functionality. The basic functionality of Lars Fennerberg's RADVD 0.7.1 is to send and receive RA messages on a router. In our implementation, the basic functionality of Lars Fennerberg's RADVD 0.7.1 is extended to satisfy the functional requirements of a GW and an MN. For a GW, RADVD is extended to transfer MAP options to CRC OLSR6, which is called a RADVD-CRC-gw. For an MN, RADVD is modified to send local RA messages to the Linux kernel by a local interface, which is called a RADVD-CRC-mn.

Our implementation is built on two main components (MN and GW) and embedded into three application modules as well as the Linux kernel. The architecture of OLSR with HMIPv6 implementation is shown in Figure 5.1 and Figure 5.2. Figure 5.1 gives the architecture of the GW implementation, and Figure 5.2 gives the architecture of the MN implementation. On a GW, the RADVD-CRC-gw module receives a normal RA with

MAP option sent by a MAP server and then sends the MAP option to the CRC OLSR6 module. The MAP option and GW network address are put into an ACM message which is transferred to an MN by OLSR routing. On an MN, when the CRC OLSR6 module receives an ACM message, it transfers the ACM message to RADVD-CRC-mn. The RADVD-CRC-mn module transfers the ACM message to a local RA message and sends it into the Linux kernel. The kernel IPv6 and HMIPv6 receive the local RA on a local interface.

Figure 5.1: Architecture of GW implementation

Figure 5.2: Architecture of MN implementation

5.3 Communication between user space and kernel space

Since our OLSR with HMIPv6 implementation exists in the user space and Linux kernel space, it is necessary to achieve data exchange between the user space and kernel space. In Linux v2.4.18-3, some specific system calls provide communication between the user space and kernel space. A system call can pass some parameters to the kernel as well as get some data from the kernel space. However, the existing Linux system calls do not provide the specific support for data exchange needed between the user space and kernel space in our implementation. Two alternatives are available to support the specific needs for the communication between user space and kernel space in our implementation: creating a new system call or extending an existing system call. Extending an existing system call is easier for some uncomplicated tasks. Since our implementation only requests simple data transfer from user space to kernel space or from kernel space to user space, an existing system call `ioctl()` is extended to satisfy our needs.

5.3.1 Communication from user space to kernel space

As discussed in Section 5.2, for an MN, the kernel IPv6 and HMIPv6 receive local RAs on a local interface. The purpose of the introduction of local RAs is to transfer MAP options and network prefix information in ACM messages into the kernel IPv6 and HMIPv6. There is a problem with the local interface and the normal operation of IPv6 and HMIPv6. The normal operation of IPv6 and HMIPv6 on an MN is to receive RAs on the interface of a hardware device, such as “wlan0” or “eth0”. In the CoA autoconfiguration and MAP entry addition operations of IPv6 and HMIPv6, information on the hardware device is needed. In our scheme of integrating OLSR with HMIPv6, the ACM take the same service as the RA messages. On an MN, the ACM messages are received as UDP packets on the interface of a hardware device. However, the kernel IPv6 and HMIPv6 cannot obtain information on the hardware device while receiving the local RAs on a local interface. Therefore, the kernel IPv6 and HMIPv6 need to be told the information on the hardware device receiving ACM messages by other means.

To achieve this purpose, we use a mechanism to transfer the name of the device receiving ACM messages from CRC OLSR6 to the kernel IPv6 and HMIPv6. This communication mechanism from the user space to kernel space is shown in Figure 5.3. In the mechanism, the device name is provided by CRC OLSR6 and passed into kernel space through an new ioctl() entry “SIOCSACM”. In the Linux kernel, the `__dev_get_by_name` function is exploited to classify the hardware device by the device name. A new function `ndisc_devname_set` is added in the kernel IPv6. The CRC OLSR6 in user space opens a raw socket and invokes the system call `ioctl(SIOCSACM)` to pass the device name to the Linux kernel while initializing.

Figure 5.3: Communication from user space to kernel space on MN

5.3.2 Communication from kernel space to user space

On an MN, CRC OLSR6 supports routing within a MANET by calculating and adding new route entries into the kernel route table, which is achieved by using a system call “add route” in Linux. However, the operation should not be done until HMIPv6 in kernel space has cleaned the old route entries in the route table. Therefore, to keep the synchronization between the OLSR6 operation and HMIPv6 operation, HMIPv6 must send an inform message to OLSR6 when an old route table is cleaned.

The detailed MN implementation is described in Section 5.5. In this section, we discuss the implementation of the inform message. As the inform message is sent from the Linux kernel space to user space, the communication from the kernel space to user space needs to be considered. This can be achieved using two different models: one is to poll the kernel space from the user space, and another is to write a driver in the kernel space to

manage an interrupt. The disadvantage of the polling model is that the whole user process polling the kernel has to wait until the message is received from the kernel space. However, the polling model is a simple way for a user process to get data from the kernel. In our synchronization process, OLSR6 should not continue its operations until it receives the inform message from HMIPv6 to indicate that the route table is cleaned, so we choose the polling way to achieve the communication from the kernel space to user space.

The communication mechanism from Linux kernel space to the user space on an MN is shown in Figure 5.4. We define new flags and functions in kernel space to monitor the deletion of a route table. On the other hand, `ioctl()` system call is extended to get the inform message from the kernel space. There are two new entries added in `inet6_ioctl()`. One is `SIOCSRTDEL`, used to poll the kernel about the state of the route table, and the other is `SIOCSFLAGSET`, used to reset the flags monitoring the RT state.

Figure 5.4: Communication from kernel space to user space on an MN

5.4 IPC in user space

As discussed in Chapter 4, a MANET GW serves as an access router in a subnet. The GW is responsible for receiving RAs with MAP options from a MAP server and propagating the MAP option within the MANET. Monash's HMIPv6 implementation uses RADVD to receive MAP options and propagate MAP options in Route Advertisement messages in a subnet. However, the GW implementation in our OLSR with HMIPv6 project exploits RADVD to receive MAP options and OLSR to handle MAP options and construct ACM messages with MAP option. Since OLSR and RADVD are two different user space processes on GW, Inter-Processes Communication (IPC) between RADVD and OLSR is needed.

There are three well-known IPC techniques supported by Linux: shared memory, pipes and socket. Shared memory allows two or more processes to share a given region of memory. This is the fastest form of IPC because the data does not need to be copied between the client and server. The only trick in using shared memory is synchronizing access to a given memory region among multiple processes. Often semaphores are used to synchronize shared memory access. Pipes are the oldest form of Linux IPC. A pipe is a channel of communication between two processes. Pipes can be one-way or two-way. With a two-way pipe, both applications can read and write data. However, pipes have a limitation: they can be used only between processes that have a common ancestor. Normally a pipe is created by a parent process that calls `fork()` to create a child process, and the pipe is used between the parent and child.

Sockets are a basic mechanism for IPC on a computer system, or on different computer systems connected by local or wide area networks. A socket is an endpoint used by a

process for bi-directional communication with a socket associated with another process. UNIX domain sockets work on a single machine, so they are much more like named pipes than a network protocol.

In our implementation, RADVD-CRC-gw and CRC OLSR6 are two completely independent processes without a common ancestor. The two processes request continuous data stream exchange. So, we choose UNIX domain socket as the IPC approach here. In RADVD-CRC-gw, a UNIX socket is opened and the MAP option is sent while an RA is received from a MAP server periodically. In CRC OLSR6, a `recv_map` thread is created while initializing. The `recv_map` thread opens a UNIX socket to keep receiving the MAP options from RADVD-CRC-gw.

5.5 MN implementation

5.5.1 Architecture

In our OLSR with HMIPv6 implementation, an MN fulfills the following functions: receiving ACM messages in CRC OLSR6 and converting the ACM message to a local RA message; transmitting the local RA message from user space to Linux kernel space; in the kernel IPv6 and HMIPv6, triggering all related functionality including movement detection (local movement or regional movement), auto-configuration of RCoA and LCoA, adding MAP entry and tunneling entry, as well as sending BU messages to the MAP server and its HA and CNs; when a route table (RT) is clean, sending an inform message from kernel space to CRC OLSR6 in user space to coordinate OLSR6 operations with HMIPv6 operations.

Figure 5.5: MN software architecture

These functions can be separated into five components: ACM handler, local RA transmission, local RA receiver, IPv6&HMIPv6 function, and RT clean inform. The above components are embedded into CRC OLSR6, RADVD-CRC-mn as well as the Linux kernel IPv6 and HMIPv6. The whole software architecture is illustrated in Figure 5.5.

ACM handler is responsible for receiving ACM messages and extracting network prefix information and MAP option information from ACM messages to construct local RA messages. In a local RA message, “icmp6_ACM” bit is set as a flag to differentiate itself from a normal RA message from other routers.

Local RA transmission includes two functions. The first function is to pass the device name into kernel space by calling a new system call `ioctl(SIOCSACM, devname)`. The second function is to send local RA messages into the kernel space by using the loopback address as the destination address in RADVD-CRC-mn.

Local RA receiver receives and processes a local RA message, then triggers relevant IPv6 functions. Its detailed functionality will be described in the following section.

IPv6&HMIPv6 functions do two things: one is to combine the interface's hardware address with the subnet prefix and MAP option in the local RA to configure LCoA and RCoA. The other is to trigger all necessary functions of HMIPv6.

RT clean inform aims to send a message to user space when the route table in Linux kernel space is cleaned. The function is achieved through communication from kernel space to user space, which is illustrated in Section 5.3.2.

The optimal registration mechanism to differentiate internal CNs from external CNs, which is described in Section 4.3.4, has not been implemented due to the time limit. In the current implementation of MN, an MN arriving in a MANET registers its RCoA with all CNs, including internal CNs located in the MANET and external CNs located outside the MANET.

5.5.2 Local RA receiver

A local RA message is sent from the MN's user space to its kernel space through its loop-back address. That means the RA is received on a local interface instead of on a real hardware device.

There are some differences in the features of a local RA message and those of a normal RA message received on a real hardware device. These features are the obstacles of the succeed IPv6 and HMIPv6 operations. Firstly, they have different source address: in the IPv6 header of a local RA, the source address is the MN's IPv6 address, while the source address in the IPv6 header of a normal RA is the IPv6 address of the router sending the

RA. Secondly, they contain a different device field: a local RA is received on a local interface that does not provide the address of the hardware device receiving ACM messages, so the Linux IPv6 cannot get the hardware address from the received local RA. Thirdly, they contain a different interface index which is the only ID of each interface. In the local RA, the interface index is set to the index of the local interface. However, in a normal RA, the interface index is the index of a real hardware device receiving the RA message.

In the IPv6 and HMIPv6 implementation, the above three fields in a received RA will be used in the succeed functions. In the auto-configuration of CoAs, the hardware device is used to provide the MAC address. In adding MAP entry, the interface index is used to refer to the interface receiving the RA message, which is the interface receiving ACM message in our implementation. In dealing with a handoff, the IPv6 header source address is used to construct a new router.

Therefore, a local RA could not trigger normal functions in IPv6 and HMIPv6 like a normal RA.

To overcome these obstacles, three procedures are added to the Local RA receiver. First, the source address of the local RA message is replaced by the GW network address in the prefix option of the local RA. Second, the device field in the local RA message is set by a real hardware interface receiving ACM messages, which is obtained by the device name passed through Local RA transmission. Third, the interface index field in the message is replaced by the index of the real hardware interface. The Local RA receiver is achieved by adding extension code in Linux IPv6.

5.6 GW implementation

In our OLSR with HMIPv6 implementation, a MANET GW fulfills the following functions: receiving normal RA with MAP option from a local MAP server by RADVD-CRC-gw and transmitting it to CRC OLSR6; constructing ACM messages with MAP option and its own address; and propagating ACM messages within MANET.

The functions can be separated into four components: Normal RA handler, MAP option transmission, ACM construction, and ACM propagation. The above components are embedded into the RADVD-CRC-gw and CRC OLSR6 implementation. The whole software architecture is illustrated in Figure 5.6:

Figure 5.6: Software structure on GW

Normal RA handler is responsible for receiving normal RA messages with MAP options from a local MAP server periodically. At the same time, it extracts MAP options from the normal RA messages. The MAP options will be transferred to CRC OLSR6 by the MAP option transmission component. The MAP option transmission is an IPC operation transmitting MAP option from the RADVD-CRC-gw process to the CRC OLSR6 process, which was described in Section 5.4.

The ACM construction and ACM propagation components are put into CRC OLSR6. ACM construction uses MAP option and GW address to construct an ACM message. The

propagation of ACM messages takes advantage of the propagation mechanism of HNA built into the basic CRC OLSR6 to broadcast ACM messages within MANET.

Chapter 6 Experiments and Evaluation

In the previous chapters, we introduced the design and implementation of our approach to integrate MANETs with the Internet. To investigate the efficiency of our approach for supporting roaming across WLANs and MANETs, we design and build a testbed to conduct experiments with our implementation. Different test scenarios are designed to demonstrate the functionality of our implementation and evaluate the performance of our approach. At the same time, the results from the performance testing are also used to quantify the impact of intricate features of HMIPv6 and OLSRv6 on the handoff latency and signaling load. Furthermore, a performance comparison between our implementation and CRC OLSR with MIPv6 implementation is done. The gain from employing HMIPv6 over MIPv6 in the hybrid network of MANETs and WLANs is evaluated.

This chapter is organized in the following manner. Section 6.1 describes the testbed used to conduct the experiments. In Sections 6.2 and 6.3, the functionality tests and performance tests are illustrated, including test metrics, test cases and test results. The gain achieved by our approach is assessed in Section 6.4.

6.1 Testbed

A testbed, modeling the network architecture depicted in Figure 6.1, has been designed and built. It is composed of Linux laptops and workstations equipped with IEEE 802.11b based access points and 802.11b wireless LAN cards. Five subnets are involved in the test: the home network with an HA, a correspondent network with a CN, two wireless foreign networks with wireless access points (FN1 and FN2), and a MANET network with gateway access. In the MANET network, two laptops equipped with 802.11b

wireless card communicate with each other, and function as an MN and a MANET node. To simulate local movements and regional movements of the MN, the FN1 and the MANET are put into the same domain (Domain #1); FN2 is put into another domain (Domain #2). The local mobility in Domain #1 is managed by MAP1 and the local mobility in Domain #2 is managed by MAP2. The WAN Emulator machine is used to insert artificial WAN delays between the HMIPv6 domains and the rest of the network, using NIST Net application [35]. NIST Net has not yet been updated to support IPv6 and therefore we use IPv6 over IPv4 tunnels to route IPv6 packets through the delay emulator.

The MN and MANET node are DELL Latitude C640 laptops with Pentium 3 processor. The other machines are DELL OPTIPLEX GX260T desktops with Pentium 4 processor. The wireless PC card is Linksys WPC11 ver3.0. The wireless network access point is Linksys WAP11. The Hub is NetGear model DS108.

The operating system installed on the testbed is Linux Red Hat 7.3 with kernel 2.4.18-3. The MIPv6 function is provided by the HUT's MIPL mipv6-0.9.3[34] and Lars Fennerberg's RADVD 0.7.1. The HMIPv6 implementation used in the testbed was developed by Monash University and was improved in our implementation to cooperate with OLSR6. The RADVD 0.7.1 is modified to satisfy the need of the function of GW and MN, which includes RADVD_crc_gw and RADVD_crc_mn. The OLSR6 used for MANET routing was developed by INRIA, ported to IPv6 by CRC, and improved for the Internet access in our implementation.

Figure 6.1: OLSR with HMIPv6 testbed

6.2 Functionality test

The implementation of our proposal is built on two components (MN and GW) and embedded into three application modules (CRC OLSR6, RADVD-CRC-mn, and RADVD-CRC-gw) as well as the Linux kernel. The functionality of the implementation is to support an MN's mobility across a MANET and a WLAN within a domain or across different domains. When an MN roams into a MANET where it is multiple hops away from the GW, the MN can continue active communication with CNs. In order to demonstrate that our implementation achieves these functionalities, we make a CN ping continuously an MN roaming across a WLAN and a MANET. We check the functionality of software modules in the following ways. First, we get the interface address list on the

MN with the command *ifconfig*, and the IPv6 route table on the MN using the command *netstat -inet6 -rn*. Second, we verify the HMIPv6 functionality by observing the MIPv6 dialogs on the MN, the MAP server, the HA and the CN, using the command *mipdiag -cls*. Third, we verify the CRC OLSR6 functionality by checking the route tables on the GW and other MANET nodes.

6.2.1 Test cases

To verify the functionality of OLSR with HMIPv6, two test scenarios are developed: local handoff and regional handoff. In a local handoff functionality test case, an MN roams from a WLAN (FN1) to the MANET or vice-versa, where the WLAN (FN1) and the MANET are in the same domain. In a regional handoff functionality test case, an MN roams from a WLAN (FN2) to the MANET or vice-versa, where the WLAN (FN2) and the MANET are in different domains. In both test cases, the MN is one or two hops away from the GW when it roams into the MANET. An internal CN is located in the MANET. An external CN is located in another network outside the domain of the MN.

6.2.2 Test results

1 Local handoff functionality test

In the functionality testing of local handoff, the MN first attaches to the local router of a WLAN (FN1). The MN configures its RCoA and LCoA according to received RAs from the local router. After the 802.11b wireless card in the MN is switched from infrastructure mode to ad-hoc mode, we can make the following observation. For the MN, a new LCoA with the prefix of the GW's IPv6 address is configured and the RCoA

is kept unchanged; the default GW entry is changed to the MANET GW in the route table. For the MAP server located in the domain where the MN is visiting, a binding entry between the MN's RCoA and new LCoA is added in the binding cache of the MAP server. For the HA and the CNs, the binding entries between the MN's Home address and its RCoA do not change. For the GW and other MANET nodes, the route entries destined for the MN are added in their route tables.

As a consequence, the MN can maintain active communication with the internal CN and external CN while it roams across the WLAN and the MANET within a domain.

2 Regional handoff functionality test

In the functionality testing of regional handoff, the MN first attaches to the local router of a WLAN (FN2) in Domain #2. The MN configures its RCoA and LCoA according to received RAs from the local router. After the 802.11b wireless card in the MN is switched from infrastructure mode to ad-hoc mode, we can make the following observation. For the MN, a new RCoA with the prefix of the new MAP server's IPv6 address and a new LCoA with the prefix of the GW's IPv6 address are configured; a default GW entry is built in the route table. For the new MAP server located in Domain #1 where the MN is arriving, a binding entry between the MN's new RCoA and its new LCoA is entered in the binding cache. For the previous MAP server located in Domain #2, a binding entry between the MN's old RCoA and its new LCoA is added in the binding cache. For the HA and the CNs, old binding entries for the MN are removed, and the binding entries between the MN's Home address and its new RCoA are added. For

the GW and other MANET nodes, the route entries destined for the MN are entered in their route tables.

As a consequence, the MN can maintain active communication with the internal CN and external CN while it roams across the WLAN and the MANET in different domains.

6.3 Performance test

The performance tests focus on the handoff latency, packet loss and signaling load during a handoff between a MANET and a WLAN. The MN roams between different visited networks in a domain or in different domains. An internal CN is located in the MANET. An external CN is located in another network outside the domain of the MN. In the performance tests, the MN sends a continuous stream of fixed size UDP packets (echo request) to an internal CN or an external CN every 0.1 second. The traffic flows are monitored using Ethereal [36] to measure the performance parameters of interest.

The handoff latency is defined as the interruption time between the end of attachment to the old subnet and attachment to the new subnet. The packet loss is measured for the receiving MN as the number of echo request packets lost during a handoff. We also study signaling load introduced on the Internet due to transfer of BUs and Binding update ACKs in a handoff.

6.3.1 Test metrics

To investigate the contribution of different factors to the overall handoff latency, the overall handoff latency is decomposed into components that reflect the effects of the individual factors. The overall handoff latency can be decomposed into different parts.

The decomposition of local handoff latency is pictured in Figure 6.2. The decomposition of regional handoff latency is pictured in Figure 6.3. They are formally expressed by the following equations:

$$L_{handoff} = L_{MDS} + L_{BU,A}, \text{ where}$$

$$L_{MDS} = L_{LL} + L_{HNA} + L_{RA} + L_{OLSR}$$

$L_{handoff}$ is the overall handoff latency.

L_{MDS} is the component of latency during which an MN detects that it has moved out of its current coverage, switches to ad hoc mode, and re-establishes communication with the CNs under the ad hoc mode.

L_{LL} is the link layer delay, which refers to the latency of the mode-detection and switching component to switch the 802.11 card from infrastructure mode to ad hoc mode.

L_{ACM} is the latency for the MN in capturing the first ACM message in the ad-hoc mode subsequent to a link layer handoff from the infrastructure mode. This component of the handoff latency is a function of the inter-arrival time of ACM messages as well as the HELLO message and TC message frequencies. The hop-count between the MN and the MANET gateway, and the traffic load distribution within the MANET, may cause jitter in the ACM arrival time.

L_{RA} is the amount of time consumed by the MN to respond to the received new subnet prefix information, which consists of synchronizing two user space processes and kernel functions. In the user space, there are two concurrent processes: RADVD-CRC-mn builds the local RA message according to the ACM message and transfers the local RA to the kernel, and CRC OLSR6 is waiting for the route table (RT) clean inform from HMIPv6. In the kernel space, HMIPv6 functionality includes installing or updating the MAP

option, configuring the new LCoA and deleting the old RT. After getting an RT clean inform, the CRC OLSR6 process resets RT clean flag in kernel space, then builds the HELLO message with the new LCoA and sends it out.

L_{OLSR} is the latency in establishing the incoming MN's membership to the MANET by using its new LCoA. It is the latency for all the MANET nodes to exchange HELLO messages and update their RTs in order to accommodate the incoming MN in the MANET. This component of the latency depends on the MANET size, the hop-count from the MN to the GW, and the HELLO message frequency. On the other hand, the measured value of L_{OLSR} could be affected by the HMIPv6 BU retransmission mechanism. In HMIPv6, the retransmission time is defined by the initial binding acknowledgement timeout constant and uses an exponential backoff algorithm. When MIPv6 detects a local handoff or regional handoff, it creates the first BU instantly, but it cannot send the BU out until CRC OLSR6 builds the new RT. HMIPv6 has to retransmit the BU if it does not receive an ACK after a delay determined by the value of the binding acknowledgement timeout. Since CRC OLSR6 needs about four HELLO message interval times to build the new RT after it receives the first HELLO message, the first BU cannot be sent out and the BU will be retransmitted. So, L_{OLSR} includes the retransmission time of the BU. Furthermore, in regional handoff, L_{OLSR} is also affected by the HMIPv6 regional movement detection mechanism. In HMIPv6, the MAP list is checked and the new MAP option is found at a frequency determined by the MAP list callback time. There is an overlap between L_{OLSR} and the MAP list callback time.

$L_{BU,A}$ is the minimum latency for the MN from starting to register its new LCoA with its MAP server to receiving the echo reply from its CNs. There are large variations in the

time between local handoff and regional handoff because of different registration procedures in the two kinds of handoff. In a local handoff, this latency is the time for the MN to register its new LCoA, through a BU, to its new MAP server, and, subsequently, the latency in receiving an ACK from the MAP. This component of the handoff latency is a function of the hop-count between the new MAP server and the MN. No BU is sent to the HA and CNs.

In a regional handoff, this latency includes two parts: the first part is the latency for the MN to register its new LCoA, through a BU, to its new MAP server, and, subsequently, the latency in receiving an ACK from the new MAP server. The second part is the minimum latency for the MN to register its new RCoA to the HA and CNs, and, subsequently, the latency in receiving ACKs from the HA and CNs. This component of the handoff latency is a function of the hop-count between the HA and the MN. If the smooth handoff between two domains is applied, $L_{BU,A}$ is the latency for the MN to register its new LCoA, through a BU, to its previous MAP server. The previous MAP server can forward packets from CNs to the MN's new LCoA until these CNs receive BUs from the MN.

Note: The shaded parts in the following figures indicate the difference between local handoff process and regional handoff process.

Figure 6.2: WLAN→MANET local handoff latency decomposition

Figure 6.3: WLAN→MANET regional handoff latency decomposition

6.3.2 Test scenarios

In the performance tests, we employ three parameters to evaluate the factors that could affect the measurement results, which are hop count, ACM interval and channel delay. The hop count parameter is defined as the number of hops between an MN and the MANET GW, which is used to quantify a delay generated by the multi-hop distance from an MN to a GW. The ACM interval parameter is the transmission period of ACM messages, which is used to evaluate the effect of the broadcast frequency of prefix information on the handoff latency. The channel delay parameter is the WAN transmission delay emulated using NISTnet tool, which is used to represent the delay associated with the transmission of a control message to the remote HA or CN. The following scenarios are repeated with various parameter values. The hop count includes one and two hops. The ACM interval varies from 0.2 second, to 0.4 second, 1 second and 2 seconds. The channel delay is emulated by setting the different one-way delay in NISTnet with a value of 0.3 second, 0.6 second or 1 second.

The performance tests include the two main scenarios: local handoff from a WLAN to the MANET, and regional handoff from a WLAN to the MANET.

In the local handoff tests, the MN roams from a WLAN (FN1) to the MANET, where the WLAN and the MANET are in the same domain. The roaming scenario from a WLAN to the MANET is repeated by using different ACM and HELLO intervals. A second set of tests is performed when the MN moves two hops away from the GW. The ACM and HELLO intervals are varied accordingly.

In the regional handoff tests, the MN roams from a WLAN (FN2) in a domain (Domain #2) to the MANET in another domain (Domain #1). Similar to the local handoff case, the

roaming scenarios in this test case are repeated using different ACM intervals and different hop counts.

6.3.3 Handoff time and packet loss

While the MN does local handoff or regional handoff from a WLAN to a MANET, it maintains active communication with CNs. According to the decomposition of test metrics described in Section 6.3.1, the handoff latency, as well as every other element of latency, is recorded in our tests. The packet loss during each handoff is also measured.

Table 6.1 shows the packet loss per local handoff and the local handoff latency as well as the various components of the delay. Table 6.1 quantifies the effect of the ACM interval and hop counts on local handoff latency and packet loss. The different WAN channel delays have no effect on the local handoff latency. Table 6.2 shows the packet loss per regional handoff and the regional handoff latency as well as the various components of the delay. Table 6.2 quantifies the effect of the ACM interval, hop counts and WAN channel delay on the regional handoff latency and packet loss. The delay measurements as well as the ACM intervals are expressed in seconds. The packet loss measurements are expressed in numbers of echo reply packets.

The handoff latency and packet loss shown in Tables 6.1 and 6.2 are the average of 20 test results and the 90% confidence intervals for 20 samples. The packet loss is proportional to the handoff latency in all scenarios.

	Packet Loss (numbers of packets)	Handoff Latency (sec)	L _{LL} (sec)	L _{HNA} (sec)	L _{RA} (sec)	L _{OLSR} (sec)	L _{BU/A} (sec)	HOP cnt.	ACM Int. (sec)
Average	23	2.51	0.76	0.20	0.08	1.48	0.05	1	0.2
Conf. Int.	[22, 24]	[2.37, 2.65]	[0.72, 0.80]	[0.13, 0.27]	[0.07, 0.09]	[1.46, 1.50]	[0.03, 0.07]		
Average	27	3.04	0.79	0.70	0.09	1.50	0.06	2	
Conf. Int.	[26, 28]	[2.89, 3.19]	[0.75, 0.83]	[0.64, 0.76]	[0.07, 0.11]	[1.43, 1.57]	[0.03, 0.09]		
Average	21	2.35	0.75	0.14	0.09	1.49	0.04	1	0.4
Conf. Int.	[20, 22]	[2.23, 2.47]	[0.70, 0.80]	[0.10, 0.18]	[0.08, 0.10]	[1.48, 1.50]	[0.02, 0.06]		
Average	26	2.92	0.76	0.65	0.08	1.49	0.05	2	
Conf. Int.	[25, 27]	[2.78, 3.06]	[0.70, 0.82]	[0.53, 0.77]	[0.07, 0.09]	[1.47, 1.52]	[0.03, 0.07]		
Average	24	2.78	0.80	0.40	0.09	1.49	0.03	1	1
Conf. Int.	[23, 25]	[2.64, 2.92]	[0.70, 0.90]	[0.31, 0.49]	[0.08, 0.10]	[1.48, 1.50]	[0.02, 0.04]		
Average	31	3.58	0.83	1.13	0.10	1.50	0.06	2	
Conf. Int.	[30, 32]	[3.42, 3.74]	[0.75, 0.91]	[1.01, 1.25]	[0.09, 0.11]	[1.48, 1.52]	[0.03, 0.09]		
Average	28	3.11	0.92	0.65	0.09	1.50	0.05	1	2
Conf. Int.	[27, 29]	[2.98, 3.24]	[0.84, 1.00]	[0.53, 0.77]	[0.07, 0.11]	[1.48, 1.52]	[0.04, 0.06]		
Average	34	3.98	0.97	1.25	0.09	1.50	0.06	2	
Conf. Int.	[33, 35]	[3.78, 4.18]	[0.89, 1.05]	[1.13, 1.37]	[0.08, 0.10]	[1.45, 1.55]	[0.02, 0.10]		

Table 6.1: Local handoff latency and packet loss

	Packet Loss (numbers of packets)	Handoff Latency (sec)	L _{LL} (sec)	L _{HNA} (sec)	L _{RA} (sec)	L _{OLSR} (sec)	L _{BU/A} (sec)	Channel Delay (sec)	Hop Cnt.	ACM Int. (sec)	
Average	20	2.20	0.76	0.15	0.09	1.04	0.11	0.3	1	0.4	
Conf. Int.	[19, 21]	[2.06, 2.34]	[0.69, 0.83]	[0.11, 0.19]	[0.08, 0.10]	[0.89, 1.19]	[0.10, 0.12]				
Average	20	2.19	0.78	0.18	0.08	0.99	0.10	0.6			
Conf. Int.	[19,21]	[2.05, 2.33]	[0.70, 0.86]	[0.16, 0.20]	[0.07, 0.09]	[0.89, 1.09]	[0.08, 0.12]				
Average	26	2.98	0.80	0.68	0.09	1.30	0.12	0.3	2		
Conf. Int.	[25, 27]	[2.82, 3.14]	[0.72, 0.88]	[0.64, 0.72]	[0.07, 0.11]	[1.10, 1.50]	[0.10, 0.14]				
Average	27	2.99	0.81	0.68	0.10	1.32	0.13	0.6			
Conf. Int.	[26, 28]	[2.83, 3.15]	[0.73, 0.89]	[0.64, 0.72]	[0.08, 0.12]	[1.12, 1.52]	[0.10, 0.16]				
Average	28	3.10	0.81	0.39	0.10	1.78	0.12	0.3	1	1	
Conf. Int.	[27, 29]	[2.96, 3.24]	[0.76, 0.86]	[0.30, 0.48]	[0.09, 0.11]	[1.58, 1.98]	[0.10, 0.14]				
Average	28	3.15	0.81	0.41	0.09	1.83	0.11	0.6			
Conf. Int.	[26, 30]	[3.00, 3.30]	[0.75, 0.87]	[0.33, 0.49]	[0.07, 0.11]	[1.72, 1.94]	[0.10, 0.12]				
Average	35	4.08	0.85	1.15	0.10	1.89	0.13	0.3	2		
Conf. Int.	[33, 37]	[3.93, 4.23]	[0.77, 0.93]	[1.02, 1.28]	[0.08, 0.12]	[1.77, 2.01]	[0.12, 0.14]				
Average	36	4.10	0.85	1.15	0.10	1.94	0.12	0.6			
Conf. Int.	[33, 39]	[3.93, 3.27]	[0.77, 0.93]	[1.02, 1.28]	[0.08, 0.12]	[1.82, 2.06]	[0.11, 0.13]				

Table 6.2: Regional handoff latency and packet loss

- **Handoff latency decomposition**

Figures 6.4 and 6.5 illustrate more straightforward images of the test results for the one hop case and two hops case respectively.

Figure 6.4: Decomposition of handoff latency (Hop cnt. = 1)

Figure 6.5: Decomposition of handoff latency (Hop cnt. = 2)

The L_{LL} delay is obtained by using the handoff_CRC application [15], which is designed to support the handoff between the two distinct 802.11 operational modes: the Ad-hoc mode and the Infrastructure mode. The delay is the time spent on the mode switch of wireless card. The delay is not affected by ACM interval and hop count.

The L_{ACM} delay indicates the time when an MN joins the MANET by using its old LCoA and receives the first ACM message. The delay increases when the MN moves into the MANET two hops from the gateway. The ACM messages have to travel through the MPRs to reach the MN. Therefore it takes more time for the control messages to reach the MN. Once the MN has received an ACM message, the HMIPv6 process configures the MN's LCoA. The MN begins to exchange HELLO messages with its neighbors using its new LCoA. With information obtained from the HELLO messages, the nodes in the MANET recalculate their route table and include the MN's LCoA as a route entry. After the HMIPv6 process configures the new LCoA, it sends its first BU to the MAP immediately, even though the OLSRv6 process has not finished building its new route table. As a consequence, the HMIPv6 process will retransmit the BU because it will not receive an ACK from the MAP before its timer expires. The L_{OLSR} delay shown in Figures 6.4 and 6.5 includes the retransmission time of the BU.

The L_{RA} delay is invariant over the different ACM intervals and different hop counts. This delay is the same value in both local handoff case and regional handoff case.

The $L_{BU/A}$ delay depends mainly on the transmission time for the channel. It is not affected by the change of the interval time of OLSR6 control messages, nor by the hop count.

- **Handoff latency vs. ACM interval**

The handoff latencies for the one hop case and two hops case are shown in Figure 6.6. Under each case, the frequency of ACM messages varies from 0.2, to 0.4, 1, and 2 seconds. Accordingly, the frequency of HELLO varies from 0.05, to 0.1, 0.25, and 0.5 seconds. The differing handoff latencies under various ACM intervals are shown in Figure 6.6.

Figure 6.6: Handoff latency vs ACM interval

Obviously, as the hop count increases, the handoff latency increases because the MN needs more time to join the MANET, to get the complete topological information about the MANET, and to update its route table.

Clearly, improvements in the handoff latency can be achieved at the expense of bandwidth by reducing inter-arrival periods of ACM and HELLO messages. However, if the inter-arrival period of ACM and HELLO messages is reduced to a very small value, it leads to excessive collisions and/or channel busy conditions as defined in IEEE 802.11b Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, resulting

in an increase in the handoff latency [33]. When the inter-arrival values are large it takes a longer time for the MN to receive the HELLO and ACM messages. As a consequence, it takes more time before the MN can calculate its new LCoA and update the route table. This tradeoff could have more effect on the handoff latency when the number of MNs and the size of the MANET increase. Therefore, an optimum value exists somewhere in between the lowest point and highest point, for a given traffic load condition.

- **Handoff latency vs. WAN channel delay**

The handoff latencies for different WAN channel delays under the one hop case and two hops case are shown in Figure 6.7 and Figure 6.8 respectively.

Figure 6.7: Handoff latency vs. WAN channel delay (Hop cnt. = 1)

Figure 6.8: Handoff latency vs. WAN channel delay (Hop cnt. = 2)

Under each case, the WAN channel delay varies from 0.3s to 0.6s and 1s. In the two figures, the test results of CRC OLSR with MIPv6, OLSR with HMIPv6 local handoff and regional handoff for different WAN channel delays are compared.

From Figure 6.7 and 6.8, we can make the following observations. The performance of local handoffs does not vary as the WAN channel delay increases. This is a consequence of the independence of $L_{BU/A}$ delay, because the transfer of BU/ACK is limited in the local domain and does not go through the WAN. The performance of regional handoffs is not affected by the WAN channel delay because of the use of smooth handoff between the previous MAP and the new MAP in Monash's HMIPv6 implementation. The delay for regional handoff includes the time spent in the encapsulation and forwarding processes, which are taken by the previous MAP and the new MAP. For the CRC OLSR with MIPv6, the increase in WAN channel delay impacts on the performance of the handoffs.

6.3.4 Signaling load

One of the most important criteria that affect the scalability property of a mobility management scheme is its signaling load. The signaling load is measured using the number of BUs and ACKs during each handoff. In the tests of signaling load, we differentiate two types of mobility: local mobility and regional mobility. The test results are presented in Table 6.3. In this table, the test results for signaling load of CRC OLSR with MIPv6 are also presented.

		HA	CN	Previous MAP	Current MAP
OLSR & HMIPv6	Local Handoff				1 BU, 1 ACK
	Regional Handoff	1 BU, 1 ACK	1 BU, 1 ACK	1 BU, 1 ACK	1 BU, 1 ACK
OLSR & MIPv6	Handoff	1 BU, 1 ACK	1 BU, 1 ACK		

Table 6.3: Control messages sent and received by an MN per handoff

The control messages sent to and received from the current MAP server result in the local signaling load. In this evaluation, we do not consider the local signaling load since local resources are not the most critical [3]. We only quantify the signaling load introduced with our approach on the Internet backbone, which is emulated using the NISTnet tool in our testbed.

6.4 Evaluation of gain

According to the results of performance tests, in this section, we assess the gain achieved by our approach over CRC OLSR with MIPv6.

6.4.1 Gain on handoff latency

To quantify the benefit of OLSR with HMIPv6 on reducing the handoff latency, we calculate the gain achieved by our approach over CRC OLSR with MIPv6. We denote G_{LO} as the gain in handoff latency when the MN is roaming within a domain, and G_{REG} as the gain in handoff latency when the MN is roaming from one domain to another domain. G_{LO} and G_{REG} are defined as follows:

$$G_{LO} = (L_{OLSR\&MIP} - L_{LO}) / L_{OLSR\&MIP}$$

$$G_{REG} = (L_{OLSR\&MIP} - L_{REG}) / L_{OLSR\&MIP}$$

Where $L_{OLSR\&MIP}$ is the handoff latency for CRC OLSR with MIPv6 implementation, L_{LO} and L_{REG} are the local handoff latency and regional handoff latency for our implementation respectively.

G_{LO} values for different WAN channel delays computed from the test results are presented in Table 6.4. G_{REG} values for different WAN channel delays computed from the test results are presented in Table 6.5.

Channel delay \ Hop count	0.3 seconds	0.6 seconds	1 second
One hop	0.11	0.19	0.27
Two hops	0.12	0.18	0.25

Table 6.4: G_{LO} in local handoff

Channel delay \ Hop count	0.3 seconds	0.6 seconds	1 second
One hop	0	0.08	0.17
Two hops	0	0.06	0.15

Table 6.5: G_{REG} in regional handoff

These results presented in Tables 6.4 and 6.5 show that our approach achieves a gain in handoff latency over CRC OLSR with MIPv6. The G_{LO} and G_{REG} are larger for larger WAN channel delays.

6.4.2 Gain on signaling load

According to the test results of Table 6.3, we can compare the signaling load of CRC OLSR with MIPv6 with the signaling load of our approach on the Internet backbone. We denote SL_{LO} as the signaling load when the MN is roaming within a domain, SL_{REG} as the signaling load when the MN is roaming from one domain to another domain, and $SL_{OLSR\&MIP}$ as the signaling load of each handoff with CRC OLSR with MIPv6 implementation. We refer to the number of CNs as N_{CN} . The values of SL_{LO} , SL_{REG} , and $SL_{OLSR\&MIP}$ are the following:

$$\begin{aligned} SL_{OLSR\&MIP} &= 2 \times (N_{CN} + 1) \\ SL_{LO} &= 0 \\ SL_{REG} &= 2 \times (N_{CN} + 2) \end{aligned}$$

Based on the signaling load on the Internet backbone for each of these schemes, we can calculate the gain achieved by our approach over CRC OLSR with MIPv6. We denote $G_{SIG,LO}$ as the gain in signaling load when the MN is roaming within a domain, and $G_{SIG,REG}$ as the gain in signaling load when the MN is roaming from one domain to another domain. $G_{SIG,LO}$ and $G_{SIG,REG}$ are defined as follows:

$$\begin{aligned} G_{SIG,LO} &= (SL_{OLSR\&MIP} - SL_{LO}) / SL_{OLSR\&MIP} \\ G_{SIG,REG} &= (SL_{OLSR\&MIP} - SL_{REG}) / SL_{OLSR\&MIP} \end{aligned}$$

We also evaluate $G_{SIG,AV}$, the weighted average of $G_{SIG,LO}$ and $G_{SIG,REG}$. By making use of the results established in [2] that 69% of a host's mobility is local, it is defined as follows:

$$G_{SIG,AV} = 0.69 \times G_{SIG,LO} + 0.31 \times G_{SIG,REG}$$

According to the above formula, we attain the gain $G_{SIG,AV}$ for different amounts of CNs, displayed in Figure 6.9.

Figure 6.9: $G_{SIG,AV}$ for different values of N_{CN}

The gains computed from the test results show that $G_{SIG,AV}$ is a function of the amount of CNs. The average gain is always larger than 54% and converges to 69%. With the growth of the size of the network and the amount of CNs, the gain in the signaling load achieved by our approach over CRC OLSR with MIPv6 becomes larger.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

With the increasing use of MANETs, mobile users desire to roam between MANETs and other parts of the Internet. Therefore, research on the integration of MANETs and the Internet is attracting attention. A mechanism supporting the integration of MANETs and the Internet should not only provide access to the Internet for MANETs, but also manage the mobility of mobile users across MANETs and WLANs.

The work of this thesis can be examined from three perspectives. The first aspect focuses on the design of a novel scheme to integrate MANETs with the Internet, which consisted of the following two tasks.

- A) Selecting a MANET routing protocol and a micro-mobility management protocol in the integration scheme.

The table-driven MANET routing protocols, such as OLSR, reduce the route acquisition latency as compared to the on-demand routing protocols, such as AODV and DSR. Furthermore, there are two optimizations available in OLSR. First, the number of control messages, which are exchanged periodically between nodes, is reduced due to the use of MPRs. Second, the flooding of GW advertisement packets is avoided since only MPRs retransmit the broadcast messages in a MANET. All the salient features of OLSR facilitate access to the Internet and mobility management support. Therefore, we choose OLSR as the routing protocol within a MANET in our scheme.

In an environment where MNs travel frequently across MANETs and WLANs within a domain, micro-mobility management protocols have better performance than the basic

Mobile IP protocols. Compared to the basic mobility support in IPv6, the HMIPv6 protocol can reduce delays during local handoffs and the signaling load on the Internet. Therefore, we choose HMIPv6 to manage the mobility between MANETs and WLANs.

B) Extending the OLSR protocol and ICMP6 protocol to support the cooperation of OLSR and HMIPv6.

To support movement detection and CoA autoconfiguration for MNs in a MANET, we added a new message type in OLSR protocol, which is called an ACM. The ACM message is used to propagate a domain's prefix and GW address to MNs which can be multiple hops away from the GW. On the other hand, to transfer the address configuration information to the HMIPv6 part in Linux, a new ICMP6 message is introduced in our scheme, the local RA message. The ICMP6 header format is extended to differentiate the local RA message from the normal RA message.

The second aspect of this research is an implementation of the OLSR with HMIPv6 scheme on Linux. We modified and extended the basic CRC OLSR6, Monash's HMIPv6 implementation and Lars Fennerberg's RADVD 0.7.1 to support the GW and MN functionality. Our implementation is embedded into these application modules as well as the Linux kernel. To support the specific needs for communication between the user space and kernel space in our implementation, we extended the existing system call `ioctl()`. Two new `ioctl()` entries and relative functions are added in IPv6 and HMIPv6.

The third aspect of the work in this thesis focuses on the functionality verification and performance evaluation of the OLSR with HMIPv6 implementation. A testbed modeling the hybrid network architecture was designed and constructed in our work. The functionality test results confirmed that the OLSR with HMIPv6 implementation

achieved the expected purpose, i.e. to support MNs' mobility across MANETs and WLANs in a single domain or in different domains. When an MN roams into a MANET where it is multiple hops away from the GW, the MN can maintain active communication with the remote CNs.

The metrics of the performance evaluation in our research are the handoff latency, packet loss and signaling load during a handoff from a WLAN to a MANET. The performance tests have addressed two tasks: first, the impact of various factors on the handoff latency and packet loss is quantified; second, the efficiency gains of the OLSR with HMIPv6 over the CRC OLSR with MIPv6 are assessed.

In the first task, the hop count representing the distance between a MN and a GW, the inter-arrival time of OLSR control messages and the WAN channel delay are the three main parameters affecting the handoff latency and packet loss. Through the performance tests, we have come to the following conclusions:

- As the hop count increases, the handoff latency and packet loss increase because the MN needs more time to join the MANET, to get the complete topology information about the MANET, to update its route table and to access the Internet through the MANET GW.
- As the inter-arrival values of OLSR control messages increase, the handoff latency and packet loss increase because a MN needs more time to exchange HELLO messages and receive ACM messages for calculating its new CoA and for updating the route table. Clearly, the handoff latency can be improved by reducing the inter-arrival time of OLSR control messages. However, if the inter-arrival time is reduced to a very small value, excessive signaling collisions could

occur so that the handoff latency and packet loss increase. Therefore, an optimum value of the inter-arrival time exists for a given traffic load condition.

- The handoff latency and packet loss during a local handoff are independent of the WAN channel delay. This is a consequence of the local mobility management mechanism of HMIPv6. The handoff latency and packet loss during a regional handoff are not affected by the WAN channel delay because of the use of the smooth handoff between a previous MAP and a new MAP in Monash's HMIPv6 implementation. This optimization is explicitly suggested in the HMIPv6 draft [3].

In the second task, the benefits of the use of HMIPv6 in our scheme are assessed. The gains from our scheme over CRC OLSR with MIPv6 on reducing handoff latency and signaling load are calculated. Regarding the reduction of local handoff latency, we found that our scheme achieves at least an 11% gain when the WAN channel delay is 0.3 second. As the WAN channel delay increases, the gain increases to up to 27% when the channel delay is 1 second. Regarding the reduction of regional handoff latency, our scheme achieves a 17% gain when the WAN channel delay is 1 second.

The average gain in reducing signaling load is a function of the number of corresponding nodes. Concerning the reduction of signaling load on the Internet, the average gain achieved by our scheme is at least 54% and converges to 69%. From the significant gain attained by our scheme in reducing signaling load, we conclude that our scheme improves the scalability of CRC OLSR with MIPv6.

7.2 Future work

The work of this thesis could be extended in the following areas: firstly, by implementing different registration operations to internal CNs and external CNs; Secondly, by investigating the effect of the handoff on some real applications; Thirdly, by combining other MANET routing protocols with HMIPv6 in such a hybrid network of MANETs and the Internet.

In the OLSR with HMIPv6 scheme, we designed an optimal registration mechanism to differentiate internal CNs from external CNs, which is described in Section 4.3.4. The main idea in the scheme is to register different types of CoAs to the two different kinds of CNs. An MN arriving in a MANET registers its LCoA with internal CNs located in the MANET, while it registers its RCoA with external CNs located outside the MANET. This mechanism can solve the bottleneck problem in the connection between external CNs and the MN without the expense of the extra indirection required for data transfer from internal CNs to the MN. To implement this optimal registration mechanism, the HMIPv6 registration process needs to be extended and some modification of Monash's HMIPv6 implementation is necessary. An internal CN list needs to be built on the MN. We suggest that the internal CN list be constructed in OLSR and transferred to the kernel. In the performance testing of this thesis, echo request and echo reply packets are used to investigate the handoff latency and packet loss during a handoff. In the future testing work, we can employ voice or video data to investigate the effect of the handoff process on the actual application in such a hybrid network.

One highlight of this thesis is to employ HMIPv6 protocol instead of Mobile IP or Mobility support in IPv6 to support mobility across MANETs and WLANs. A table-

driven routing protocol, OLSR, was used to support routing within a MANET in the integration scheme. To investigate the efficiency of the use of HMIPv6 in such a hybrid network of MANETs and the Internet, in the future work, we can combine the HMIPv6 protocol with other MANET routing protocols (such as DSR, AODV) in the hybrid network. By comparing the performance of this combination with that of existing integration mechanisms, we can provide more evidence regarding the efficiency of HMIPv6 in the integration of MANETs and the Internet.

References

- [1] A. T. Campbell et al., "Comparison of IP Micro-mobility Protocols", in *IEEE Wireless Communications Magazine*, Vol. 9, Feb. 2002.
- [2] C. Castelluccia, "INRIA Hierarchical Mobile IPv6 Proposal", INRIA, <http://www.inrialpes.fr/planete/people/bellier/hmip.html>, Jul. 2000.
- [3] C. Castelluccia, "A Hierarchical Mobility Management Scheme for IPv6", in *Proceedings of the Third IEEE Symposium on Computers and Communications*, Nov. 1998.
- [4] G. Deley, "Hierarchical Mobile Ipv6 Research", Monash University, <http://www.ctie.monash.edu.au/ipv6/hmipv6.htm>, Aug. 2002.
- [5] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", IETF RFC 1883, Dec. 1995.
- [6] M. Ergen and A. Puri, "MEWLANA-Mobile IP Enriched Wireless Local Area Network Architecture", Mobile Ad hoc conference, Paris, 2002.
- [7] E. Gustafsson, A. Jonsson and C. E. Perkins, "Mobile IP Regional Registration", Internet Draft, draft-ietf-mobileip-reg-tunnel-03, Work in Progress, Jul. 2000.
- [8] C. Hedrick, "Routing Information Protocol", RFC 1058, <http://www.faqs.org/rfcs/rfc1058.html>, Jun. 1988.
- [9] P. Jacquet et al., "Optimized Link State Routing Protocol", draft-ietf-manet-olsr-05.txt, Aug. 2001.
- [10] D. Johnson and C. E. Perkins, "Mobility Support in IPv6", in *ACM Mobicom 96*, ACM, pp. 2737, Nov. 1996.

- [11] U. Jonsson and F. Alriksson, "MIPMANET – Mobile IP for Mobile Ad Hoc Networks", in Proc. Wrk-shp on Mobile Ad Hoc Network & Comp. (MobiHoc), (Boston, USA), 2000.
- [12] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", in Mobile Computing, 1996.
- [13] G. Kirby, "Locating the User", In *Communication International*, 1995.
- [14] H. Lei and C.E. Perkins, "Ad hoc Networking with Mobile IP", Proceedings of the 1st Workshop on Mobile Ad hoc Network and Computing (MobiHoc'00), PP. 75-85, Boston, Massachusetts, Aug. 2000.
- [15] L. Lamont, M. Wang, L. Villasenor and T. Randhawa, "An IPv6 and OLSR based Architecture for Integrating WLANs & MANETs to the Internet", in WPMC' 2002, technical sessions, Oct. 2002.
- [16] A. Myles and D. Skellern, "Comparing Four IP Based Mobile Node Protocols", in *Proc. 4th Joint European Networking Conference*, 1993.
- [17] D. A. Maltz. "The Effects of On-Demand Behavior in Routing Protocols for Ad Hoc Networks", in IEEE JSAC, vol. 17, no. 8, pp. 1439-53, Aug. 1999.
- [18] D. A. Maltz and J. Broch. "Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed", IEEE Personal Communications, Feb. 2001.
- [19] S. Murthy and J. J. Garcia-Luna_Aceves, "An Efficient Routing Protocol for Wireless Networks", in ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996.
- [20] A. Neumann and A. Festag. "Hierarchical Mobile Ipv6 Implementation Experiences", Siemens IPv6 Workshop, TKN, TU-Berlin, Jun. 2002.

- [21] C. E. Perkins, "IP Mobility Support", IEEE RFC 2002, Oct. 1996.
- [22] C. E. Perkins, "Tutorial: Mobile Networking Through Mobile IP", in Internet Computing Online, <http://www.computer.org/internet/v2n1/perkins.htm>, 1997.
- [23] C. E. Perkins and D.B. Johnson, "Route Optimization in Mobile-IP", <ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-tunnel-07.txt>, Jul. 1996.
- [24] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", in Comp. Communication Rev., Oct. 1994.
- [25] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing", in Proc. 2nd IEEE Wksp. Mobile Comp. Sys. And Apps. Feb. 1999.
- [26] C. E. Perkins and P. Bhagwat. "Routing over Multi-hop Wireless Network of Mobile Computers", SIGCOMM 94 : Computer communication Review 24: 234-244, Oct. 1994.
- [27] E. M. Royer, S. Barbara and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, Apr. 1999.
- [28] R. Ramjee, T. La Porta, S. Thuel, K. Varadhan and S.Y. Wang, "HAWAII: A Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks", in Proc. IEEE International Conference on Network Protocols, 1999.
- [29] A. S. Tanenbaum, Computer Networks, 3rd ed., Ch. 5, Englewood Cliffs, NJ: Prentice Hall, pp. 357-58, 1996.

- [30] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility", in ACM SIGCOMM Computer Communication Review, vol. 29, no. 1, pp. 50-65, Jan. 1999.
- [31] R. Wakikawa, J. T. Malinen and C. E. Perkins, "Global connectivity for IPv6 Mobile Ad Hoc Networks", Internet Draft, draft-wakikawa-manet-globalv6-02.txt, Nov. 2002.
- [32] M. Wang and L. Lamont, "MANET and Mobile IP Hybrid Testing Report", CRC Technical Report, April 2002.
- [33] Y. Youozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron Spectroscopy Studies on Magneto-optical Media and Plastic Substrate Interface", IEEE Transl. J. Magn. Japan, vol. 2, pp 740-741, Aug. 1987.
- [34] Helsinki University of Technology (HUT) Telecommunications and Multimedia Lab, "MIPL Mobile IPv6 for Linux", <http://www.mipl.mediapoli.com/>.
- [35] National Institute of Standards and Technology (NIST) "NIST Net Home Page", <http://snad.ncsl.nist.gov/itg/nistnet>.
- [36] Ethereal, "The Ethereal Network Analyzer", <http://www.ethereal.com/>.

Appendix Control Flows of OLSR with HMIPv6

Implementation

In this appendix, we introduce the control flows of our OLSR with HMIPv6 implementation. The set of software is installed on MN and MANET GW

A.1 Control flows of MN implementation

When a MN receives an ACM message by CRC OLSR6 in the user space, the MAP option and GW network address are transferred to the kernel IPv6 and HMIPv6. Then, the MN could take location detection, CoA auto-configuration and registration operations. The control flows of these procedures are shown in Figure A.1~ Figure A..

The MN functionalities are implemented in the user space and Linux kernel space. The functionalities in the user space are embedded in CRC OLSR6 and RADVD-CRC-mn.

The control flow in user space is shown in Figure 5.6.

Figure A.1: Control Flow in User Space on MN

The CRC OLSR6 is extended to achieve the function of ACM handler and trigger RADVD-CRC-mn execution. The ACM message and MAP option are defined in the CRC OLSR6. The control flow of the functions in CRC OLSR6 is shown in Figure A.2. The function of the local RA transmission is implemented in RADVD-CRC-mn. The control flow of the functions in RADVD-CRC-mn is shown in Figure A.3.

Figure A.2: Control Flow in CRC OLSR6 on MN

Figure A.3: Control Flow in RADVD-CRC-mn on MN

The functionalities in Linux kernel space are achieved by modifying and extending Linux IPv6 and HMIPv6. The control flow in kernel space is shown in Figure A.4. According to received RA messages, HMIPv6 do movement detection. Based on different movement mode (local movement or regional movement, HMIPv6 handles local handoff or regional handoff. A hierarchy map list is used to store MAP option information received from RA messages. A timer called `map_gc_callback` timer is used to control the period of regional movement detection.

Figure A.4: Control Flow in Kernel Space on MN

The control flow to handle local handoff and regional handoff are shown in Figure A.5 and Figure A.6 respectively.

In Linux kernel, ten files are involved in the modification in order to achieve the functions of the local RA receiver and RT clean inform.

Figure A.5: Control Flow of Local Handoff on MN

Figure A.6: Control Flow of Regional Handoff on MN

A.2 Control flows of GW implementation

The MANET GW operation in RADVD-CRC-gw process and CRC OLSR6 process are shown in Figure A.7 and Figure A.8 respectively.

Figure A.7: Control Flow in RADVD-CRC-gw on GW

Figure A.8: Control Flow in CRC OLSR6 on GW